Report No. FAA-EM-81-1-VOL-2 (18) (19)
Vol. II

AD104155

LEVEL III

(6) - 135

MTR-81W16-
Vol. II

(6) UPGRADED FAA AIRFIELD CAPACITY MODEL.
VOLUME II: TECHNICAL DESCRIPTION OF REVISIONS,

(10) WILLIAM J. SWEDISH

The MITRE Corporation
McLean, Virginia 22102

(14) MTR-81W16-VOL-2

ADA104155

(11) FEB[redacted]81

(12) 161

(15) DT-FA01-81-C-10001

Prepared for

U.S. DEPARTMENT OF TRANSPORTATION
FEDERAL AVIATION ADMINISTRATION
OFFICE OF SYSTEMS ENGINEERING MANAGEMENT
Washington, D.C. 20591

409990

81 8 08 037

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FAA-EM-81-1, Volume II	AD-A104155	

4. Title and Subtitle	5. Report Date
Upgraded FAA Airfield Capacity Model Volume II: Technical Description of Revisions	February 1981
	6. Performing Organization Code
	8. Performing Organization Report No. MTR-81W16, Volume II
7. Author(s) William J. Swedish	
9. Performing Organization Name and Address The MITRE Corporation Metrek Division 1820 Dolley Madison Blvd. McLean, VA 22102	10. Work Unit No. (TRAIS) 11. Contract or Grant No. DTFA01-81-C-10001 13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address Department of Transportation Federal Aviation Administration Office of Systems Engineering Management Washington, DC 20591	14. Sponsoring Agency Code AEM-100

15. Supplementary Notes

16. Abstract

The FAA Airfield Capacity Model, a computer program designed to quickly calculate the runway capacity of an airport, has recently been upgraded.

Several new features have been implemented in the upgraded version. Among these are improved input and output formats for easier usage, the capability to compute runway capacity for up to eleven different percentages of arrivals in a single run (as opposed to a separate run for each percentage), and provisions for calculating the capacity of alternating arrivals to a pair of parallel runways. Several other runway configurations have been added to the model, or improved, as well.

Other changes have been made to the internal logic of the model which will result in reduced running times and/or improved accuracy. The resulting capacities may, therefore, differ from the results obtained with the previous version. In most cases this will not affect the ranking of the potential airfield changes under evaluation.

This report documents the upgraded FAA Airfield Capacity Model. Volume I, "Supplemental User's Guide," provides a general overview of the major changes that have been made to the program and includes revised versions of the relevant chapters in the existing User's Manual, FAA-RD-128.

Volume II is a detailed technical description of the revisions to the program, including flow charts of the logic and evaluations of various altenative logics. This volume is intended as a programmer's guide, but it may also be useful for the experienced analyst who desires a fuller understanding of the model.

17. Key Words	18. Distribution Statement
Runway Capacity Airport Operations Computer Models	Document is available to the public through the National Technical Information Service, Springfield, VA 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified		

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

ACKNOWLEDGEMENTS

The contribution of Frank A. Amodeo to the work described in this report needs to be recognized at the outset. Mr. Amodeo is responsible for major contributions in three areas: gap stretching, f.e.d. mix, and the convolved probabilities in the Q-logic. Mr. Amodeo developed the arrival gap stretching concepts which he describes in Appendix B, wrote the programs and performed extensive testing. He also developed and implemented the logic for calculating the first-enqueued-departure mix, and structured and derived the convolution of probabilities presented in Appendix C. For this, and for his valuable comments and consultations on matters mathematical and operational, many thanks.

The author also wishes to acknowledge the contributions to this work of James R. McGinnis, who assisted with the design, coding and testing of many of these program changes, and who suffered the many headaches of transferring the final program to the CDC time-sharing system.

EXECUTIVE SUMMARY

This report describes the recent upgrading of the FAA Airfield Capacity Model. A careful review of the original program revealed areas where changes were needed to bring the program up-to-date, to add worthwhile new capabilities, and to correct logic errors.

Among the changes described herein are the following:

- o Use of selective stretching of arrival gaps to increase departure capacity, which improves the accuracy of the capacity calculations at less-than-peak arrival rates
- o Consideration of the "first enqueued departure" mix as distinct from the overall fleet mix, in order to correct distortions in the departure capacity caused by different aircraft type characteristics
- o The ability to specify more than one arrival percentage in a given run, which can substantially reduce the cost of a complete capacity analysis
- o Calculation of capacity of alternating arrivals to parallel runways, a procedure not in use when the program was first written
- o Adjustment of the decomposition of complex configurations into one or more simpler configurations, for improved accuracy.

Details of the modifications are described, comparisons are made between the original and the upgraded versions, and in some cases, the reasons for not implementing a proposed modification are explained.

The program modifications described in this report, extensive as they may seem, are only a portion of all the improvements which were made to the capacity program. Other changes were made to reduce the program running time, decrease the storage requirements, and increase the usability of the program and the accuracy of the results. In addition, many comment statements were added, and the program input and output were modified.

The result of all these modifications is a greatly improved and more reliable program, easier for the first-time user to deal with but also with more options available to the experienced user.

The overall effect of the changes that have been made may be judged by comparing the results obtained from both the original and revised versions of the program. One such comparison is made in Table A. Capacities were calculated for Miami International Airport for both visual and instrument conditions (VMC and IMC), using input data from Reference 5, the Airport Capacity Task Force report. The revised program calculated a capacity which was 11% higher in VMC, and 2% higher in IMC, than the original program. As can be seen from the capacity curves of these two cases (Figure A), most of the increase comes not from a higher arrival-priority capacity, but from the use of intermediate capacity points. It is also worth noting that the results from the revised program show a relative difference between VMC and IMC capacities that is in closer agreement with current-day experience than the results of the original version.

It must be realized that this model will never be perfect. Changes to ATC procedures, or to predictions of future ATC procedures, will require appropriate updating of the program. New features and capabilities will be desired. Experience with the revised program may reveal previously hidden errors which need to be corrected or areas where the program logic can be improved further.

The Airfield Capacity Model should therefore be reviewed periodically in the future, to keep it current and useful.

TABLE A

COMPARISON OF MIA RESULTS, ORIGINAL AND REVISED PROGRAM VERSIONS

		ORIGINAL VERSION	REVISED VERSION	CHANGE
VMC	CAPACITY	100.9	112.3	+11.3%
	COST	12.7 CPUs	14.5 CPUs	+14.2%
IMC	CAPACITY	99.9	102.2	+ 2.2%
	COST	14.2 CPUs	14.5 CPUs	+ 2.1%

-- MIAMI TODAY

-- FAR-SPACED PARALLELS, MIXED OPERATIONS ON BOTH

-- CAPACITY AT 50% ARRIVALS

-- 1 CPU second = $.20 (MITRE IBM 370/148)

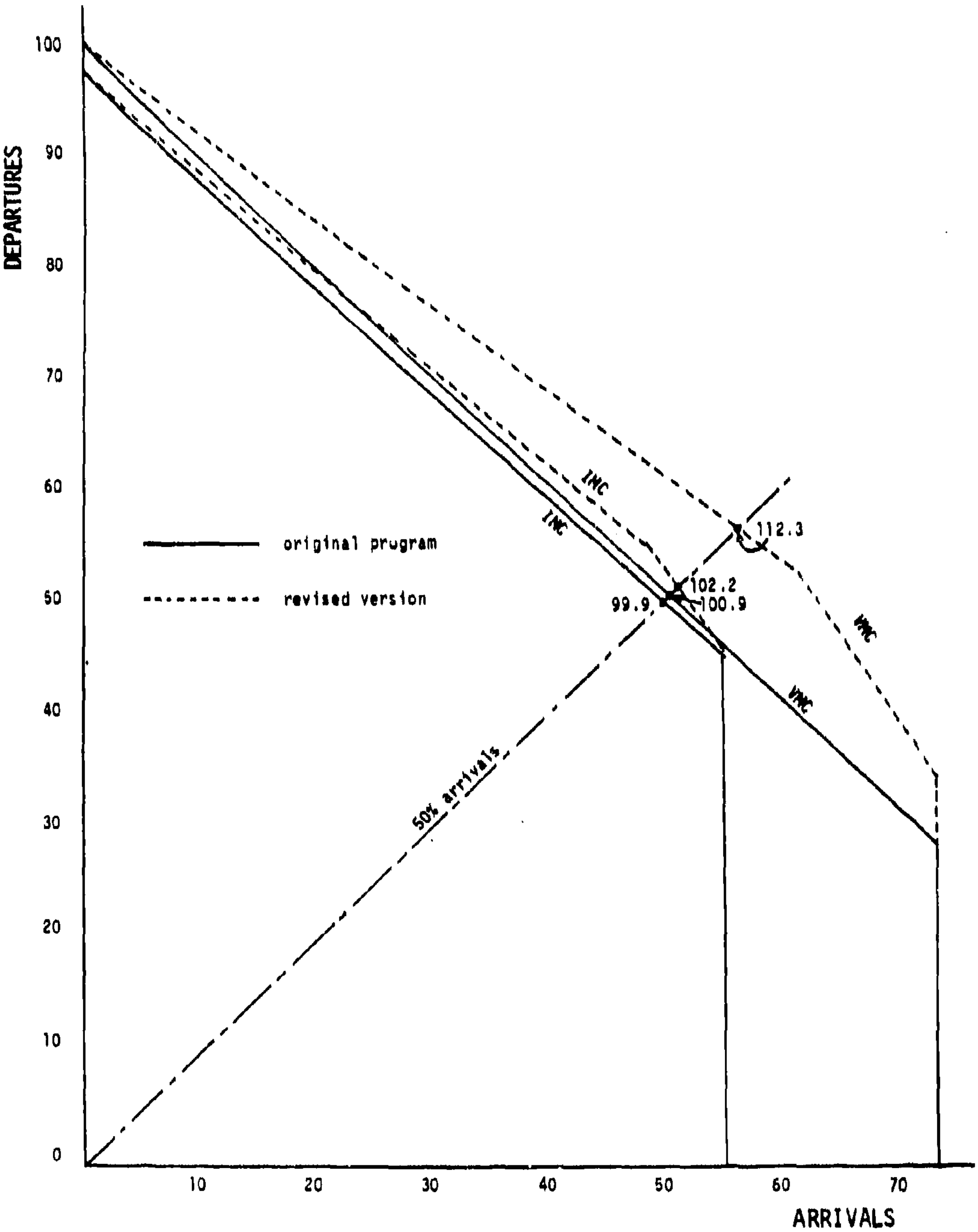


FIGURE A
MIAMI CAPACITY CURVES, ORIGINAL AND REVISED PROGRAM VERSIONS

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Concluded)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

FIGURES

LIST OF ILLUSTRATIONS
(Continued)

LIST OF ILLUSTRATIONS
(Concluded)

1. INTRODUCTION

1.1 Background

The FAA Airfield Capacity Model is a computer program that calculates the maximum operational capacity of a runway system. It is an analytical model, as opposed to a simulation. The model user has considerable freedom to vary the parameters of the computation, such as number and usage of runways, aircraft mix and speeds, and the characteristics of the ATC system.

The capacity model was originally developed for the FAA in the early 1970s by a consortium which included Peat, Marwick, Mitchell and Company (PMM&Co.) and McDonnell Douglas Automation (MCAUTO). The program was further modified by the FAA's Systems Research and Development Service (SRDS). The model has been used by the FAA for the Airport Capacity and Delay Task Force studies, and is currently available to the public through the Control Data Corporation (CDC) timesharing computer service.

Use of the model during the first phase of the Airport Capacity and Delay Task Forces raised certain questions about model assumptions and capabilities. Other attempted uses of the capacity program, in connection with various delay studies as well as the O'Hare Configuration Management model, pointed out other shortcomings of the program. A detailed appraisal of the program itself was then undertaken.

This review identified three basic areas where modifying the program would be worthwhile:

- o Adding new functions and abilities
- o Incorporating changes to the ATC procedures
- o Correcting errors in the program code and logic.

Volume I of this report provides an overview of the program modifications, focusing on changes to the input and output. It is intended for the general user. The present document, Volume II, will describe in detail the changes which were made to the program, the reasons why the changes were made, and the effect of the changes on the results and cost of running the program, for the benefit of the programmer or experienced analyst.

1.2 Description of the Airfield Capacity Model

1.2.1 Basic Concepts

In the program, "capacity" is defined as the maximum sustainable runway throughput, on a long-term basis, of arrivals and departures given a continuous sustained demand. It is a theoretical number based on average aircraft mixes and average controller performance. Actual throughputs obtained in the field may be different, either higher or lower, due to differences in aircraft mix, velocity profiles, controller performance, etc., from what is assumed by the program. Nevertheless, this theoretical capacity is valid for making comparisons between airports or for gauging the effect of changes to the ATC system or procedures, or to the airport environment.

The capacity of a departures-only runway is found by first determining the average time between departures, then inverting this to obtain the departure rate in terms of operations per hour. The capacity of an arrivals-only runway is calculated in a similar manner. The average time between arrivals at the runway threshold is a function of the required separation between the arrivals, their velocities, the length of the common path, the runway occupancy time of the lead aircraft, and the mix of aircraft types.

The capacity of a runway with mixed operations depends upon the desired ratio of arrivals to departures. To a certain extent, arrivals and departures can be traded off, but it is rarely possible to do this on a one-to-one basis. To obtain the mixed capacity, the following steps are taken.

First, the "arrival-priority" capacity is calculated (see Figure 1-1). The arrival capacity (A1) is obtained as above, with no consideration given to departure requirements. The expected number of departures which can be released in each arrival gap without disrupting the arrivals is then calculated, based upon probable distributions of interarrival times and runway occupancies. The resulting departure capacity is D1 in Figure 1-1.

The program then compares the proportion of arrivals so obtained (A1/(A1 + D1)) with the desired arrival percentage as input.* If the desired percentage is higher, excess departures are dropped (the curve below (A1, D1)). If the desired percentage is lower, a new set of capacity values is calculated for the same configuration

* The user inputs this number as a percentage (e.g., 50), but it is converted within the program to a proportion (0.50). In this report, we shall assume that this conversion is understood.

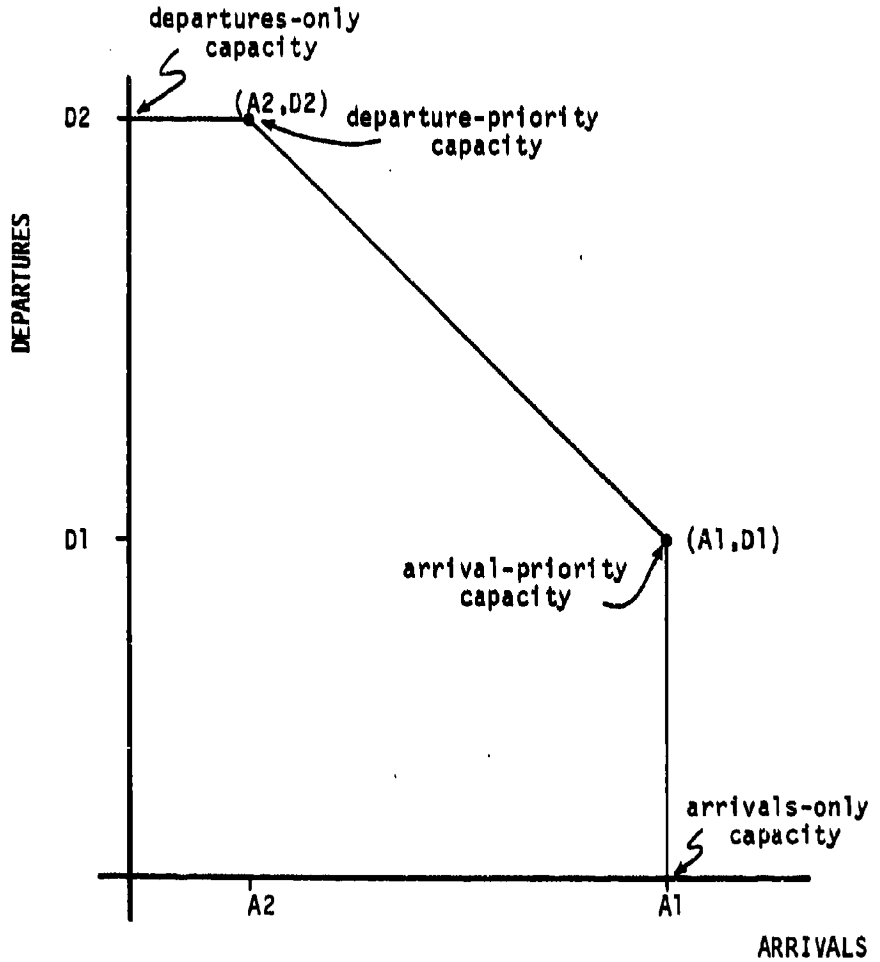


FIGURE 1-1
REPRESENTATIVE CAPACITY CURVE

with all conflicting arrival streams removed. For example, for a single arrival-departure runway the capacity of the runway is computed for departures only. This "departure-priority" capacity consists of an arrival capacity A2 and a departure capacity D2. If the desired percentage of arrivals is less than A2/(A2 + D2), excess arrivals are dropped (the curve to the left of (A2, D2)).

Otherwise, the program linearly interpolates between (A1, D1) and (A2, D2). This represents operations for part of the hour in each configuration, approximating the stretching of arrival gaps to allow additional departures.

1.2.2 Terminology

Some of the other terminology associated with the program should also be explained at this time. For example, a runway "configuration" is a unique runway layout, with a specified number and arrangement of runways and with the arrivals and departures assigned to particular runways. A "model" is the subsection of logic in the program which is unique for each configuration. A "case" is a set of input data for which the capacity is to be calculated; a "run" can consist of several cases in which one or more pieces of input data is changed in each case.

The configuration to be evaluated by the program is specified by inputting the number of the appropriate model. Each model number consists of two parts, the general model series and the particular operating strategy. For example, model 1-3 is a single runway (series 1), operating strategy 3, which is mixed operations (also indicated B, or both). There are thirteen model series, including 2 (two parallel runways), 3 (three parallels), and 6 (two intersecting). These series are shown in Figure 1-2. The strategy codes are unique for each series. Each model will be referred to in this report in a shorthand form, e.g., model 2-20 (C:A,D). C indicates close-spaced, A means arrivals on runway 1, D means departures on runway 2. The symbols used will be:

C -- close spaced parallels (700-2499 ft)

N -- near-spaced (2500-3499 ft)

M -- medium-spaced (3500-4300 ft)

F -- far-spaced (4300 and above)

A -- arrivals

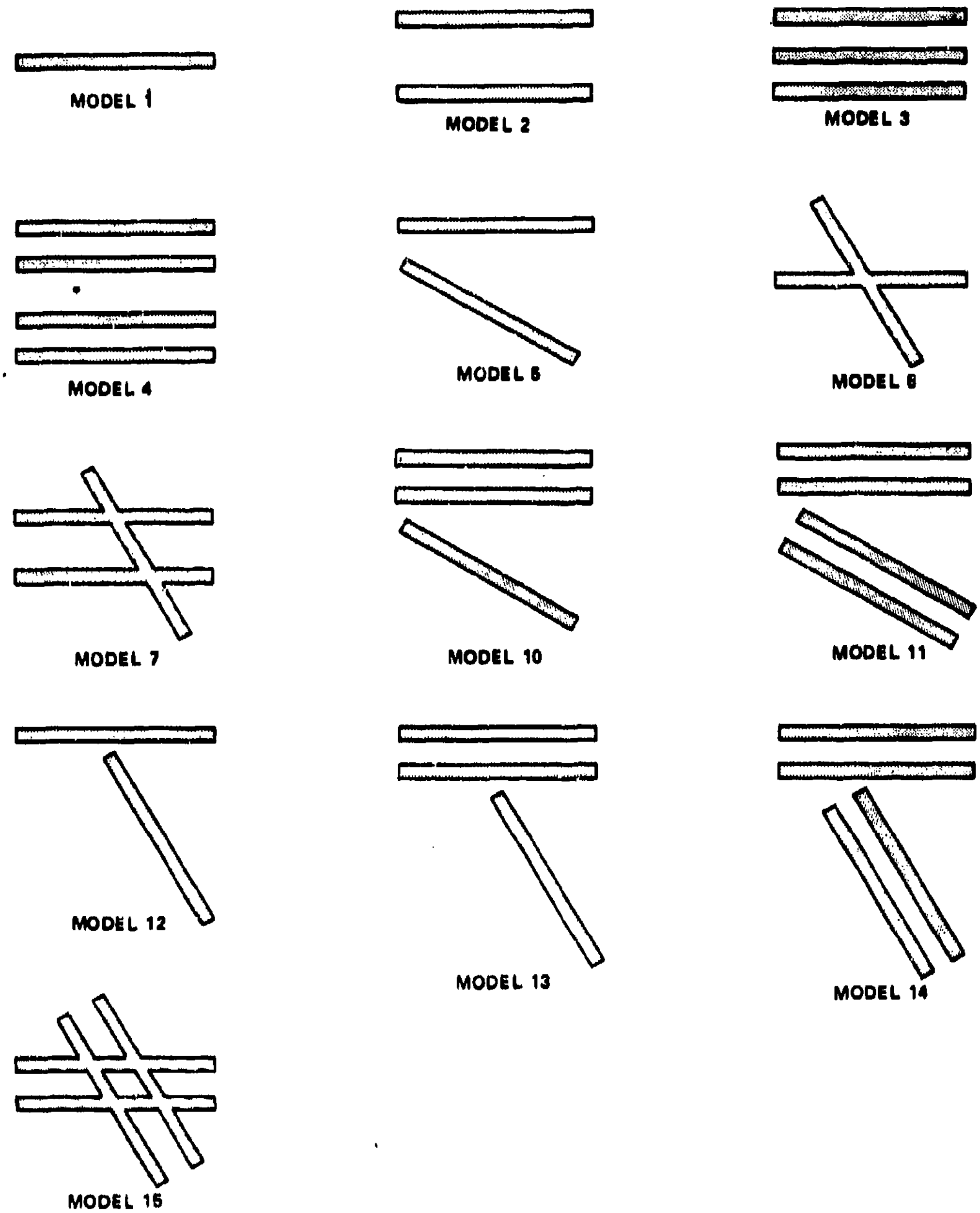


FIGURE 1-2
MODEL SERIES AND RUNWAY CONFIGURATIONS

D -- departures

B -- both

A glossary of all symbols, abbreviations and variable names used in this report may be found as Appendix A.

If the operational symbols A, D, and B do not follow the natural runway sequence, this will be indicated. For instance, (C:A2,D3) would indicate arrivals on runway 2 and departures on runway 3.

1.2.3 Weather Categories

The capacity program recognizes three general weather categories:

- VMC -- Visual Meteorological Conditions
- MMC -- Marginal Meteorological Conditions
- IMC -- Instrument Meteorological Conditions

The following are the main functional differences between these three categories.

VMC exists when ceilings are greater than or equal to 1000 feet, and visibility is greater than or equal to 3.0 miles. Visual Flight Rules (VFR) apply. A departure can be released if it will clear the runway before the arrival crosses the threshold. Arrivals and/or departures on one runway may be conducted independently of operations on a parallel runway, except for separations required by wake vortices, if the runways are 700 feet or more apart. Visual separations are applied between arrivals.

Below 1000 foot ceilings and 3.0 mile visibilities, Instrument Flight Rules (IFR) are enforced. Radar separations apply between consecutive arrivals. Operations on parallel runways are dependent to the following extent:

- Arrival/arrival (AA) operations are dependent if the runways are separated by less than 4300 feet. Alternating arrivals with a 2.0 nmi diagonal are allowed if the separation is 3000 feet or more. Below 3000 feet, IFR longitudinal separations apply.
- Departure/departure (DD) operations are dependent if the runway separation is less than 2500 feet.

- o As in VMC, AA and DD operations are vortex-dependent if the runway separation is less than 2500 feet. This means that separations larger than 3.0 nmi are applied between arrivals and more than 60s between departures.

These procedures are based upon the current version of the Controller's Handbook (Reference 1). Some changes have been made to these procedures since the program was originally developed.

These procedures apply to both MMC and IMC. The difference between the two derives from the treatment of departure/arrival (DA) interactions. IFR procedures call for a 2.0 nmi DA separation (termed DLTADA in the program). This is usually interpreted as a requirement that the arrival must be 2.0 nmi or further from the threshold in order to release a departure on the same or a dependent parallel runway. The Airport Capacity Program assumes that if the arrival can see the departure, visual separation can be applied between them, and the 2.0 nmi DA rule is superceded.

In MMC, marginal IFR weather, conditions are worse than VMC but visibility is better than 2.0 nmi. Visual separations can therefore be applied at all times between arrivals and departures. Requirements for DA operations are therefore the same as in VMC, although otherwise IFR rules apply.

In IMC, visibility is less than 2.0 nmi. Parallel arrivals and departures are dependent if the runways are less than 2500 feet apart. Departures cannot be released when the arrival is within DLTADA of the threshold. However, the departure can be released if the arrival is within visual range (termed EPSILN in the program). EPSILN is defined to be either the visibility or the distance at which the arrival first descends below the ceiling, whichever is less.

The distinction between MMC and IMC is only made for configurations involving both arrivals and departures. The logic for deciding whether VMC, MMC or IMC prevails may be found in Section 5.1. An example of the importance of DLTADA and EPSILN may be found in Section 3.2.

1.2.4 Program Versions

When this review of the capacity program began, two different versions of the program existed. One was the official FAA version, the other was one which PMM&Co. had retained and modified for its own use. The two versions are generally similar and the above description applies to both. There are differences, however. The

existing FAA version included some changes made by SRDS, while the PMM&Co. version did not have some of the errors found in the FAA version. The upgraded version of the capacity program has been based upon the FAA version, which will also be referred to as the "original" version in this report. The PMM&Co. version was used to help track down the causes of some errors in the FAA version.

Additional program details may be found in References 2 through 4.

1.3 Structure of This Report

Several major modifications were made to the capacity program. A method for stretching arrival gaps in order to accommodate the greatest number of additional departures will be discussed in Section 2 of this report. Section 3 will detail two other major changes which affect the calculation of the departure capacity: the "first-enqueued-departure" mix and revisions to the departure-departure separation logic.

New model capabilities are discussed in Section 4. These include the ability to specify several arrival percentages in a single case, new logic for alternating arrivals to parallel runways, and a new intersecting runway model.

Section 5 describes some of the other changes made to the program, to correct some programming errors and to update the internal ATC procedures. The work done on the model, and the need for future reviews of the capacity program, will be summarized in Section 6. Additional technical details are presented in the appendices.

2. MAJOR LOGIC CHANGE -- GAP STRETCHING

Of all the modifications which were made to the Airfield Capacity Model, a few deserve to be called major changes. These were substantial changes to the model logic, incorporating new ideas or new approaches, with wide-ranging impact. Included in this category are:

- A technique for selectively stretching arrival gaps, thereby increasing departure capacity, as a means of achieving a desired percentage of arrivals

- A means of adjusting the probabilities of departing aircraft types to deal more effectively with the constraints on overall departure mix

- Improvements to the logic for calculating the effect of departure-departure separations between arrival gaps.

The first of these major changes, gap stretching, will be discussed in this section; the other two will be discussed in Section 3.

2.1 The Problem of Varying the Arrival Percentage

As discussed in Section 1, the original capacity program calculated capacity at the desired percentage of arrivals by calculating an arrival-priority capacity and a departure-priority capacity, and extrapolating from these or linearly interpolating between them to achieve the desired arrival/departure ratio.

If there are more departures at the arrival-priority point than desired, the program will drop the excess departures. Since it is not possible to increase the number of arrivals (by definition, the arrival-priority point represents the maximum arrival capacity), there is no other way to achieve the desired arrival/departure ratio. Similarly, excess arrivals will be dropped if there are too many at the departure priority point. The original program logic in these areas has not been changed.

If the desired arrival percentage falls between that of the arrival priority point (i.e., A1/(A1 + D1) * 100) and the departure priority point (A2/(A2 + D2) * 100), then the original program would perform a linear interpolation between the two points. A physical interpretation of this would be that, for part of the hour, the airport was run in the arrival priority mode (minimum separation between arrivals) and in the departure priority mode for the remainder. In reality the hour is not likely to be so arbitrarily divided between the two operating modes; however, it was

felt when the model was developed that interpolation provided an acceptable approximation to actual capacity. An alternative interpretation of the interpolation procedure would be that a short interval of departure-priority time is inserted into each arrival gap, which is reasonably close to the real world practice of increasing the spacing between arrivals to allow additional departures.

This approximation tends to underestimate capacity, however. It is easy to imagine cases where only a slight gap stretch would be required to allow an additional departure, producing a higher departure capacity than the interpolation procedure would indicate. On the other hand, there are transition effects involved with switching between arrival-priority and departure-priority modes, such as delays caused by departure/departure and departure/arrival separations, which may not be fully accounted for. However, any such transition effects could be minimized by only switching once in the hour from arrival-priority to departure-priority mode, a procedure which is always possible, if not likely.

The inclusion of a gap stretching procedure which could stretch gaps efficiently and account for these transition effects was expected to be a worthwhile modification to the program.

2.2 Alternative Techniques for Gap Stretching

2.2.1 Subroutine SUPER

The original version of the FAA capacity program contained a gap stretching procedure of sorts, subroutine SUPER, added to the program by SRDS. SUPER works by systematically varying the minimum separation between arrivals (DLTAIJ) and recomputing the capacity each time. SUPER is unacceptable as a gap stretching technique, however, because it is inefficient, expensive, and less than accurate.

If the user wishes to run SUPER, the initial values of DLTAIJ are input, as well as values for the DLTAIJ increment (FINC) and the maximum value of the separation (GMAX). The 4 X 4 DLTAIJ matrix* is then condensed into 3 numbers by SUPER. The first number is the value input for a large aircraft following a large (LL); this is taken to be the separation used for a small followed by any type (S•) or a large followed by any type except a heavy (LS, LL). The

* The program allows four different aircraft types, which may be the lead or the trail aircraft of a pair. The four types are divided into three weight categories: small, large and heavy. Small aircraft are defined to be less than 12,500 pounds, while heavies are 300,000 pounds and above.

second value, for a heavy behind a heavy (HH), is also used for a heavy following a large (LH). Lastly, the value for a large behind a heavy (HL) is also used for a small following a heavy (HS). This is illustrated in Figure 2-1.

These separations are then incremented, in turn, until GMAX is reached. Three different procedures for incrementing DLTAIJ are available, depending on a user-specified parameter. Capacity is computed for each set of incremented DLTAIJs. The user can also specify whether all intermediate capacity values are to be printed out or just the maximum capacity.

It can be seen that SUPER is a brute-force approach to gap stretching, operating strictly by trial and error. It is expensive because the maximum possible capacity is unlikely to be attained unless a small increment and a large maximum separation are used, increasing the number of cases to be run. Even so, SUPER would give answers which were inaccurate if not simply wrong, because the 4 X 4 DLTAIJ matrix cannot always be characterized by only three different numbers.* In addition to the inaccuracy of the simplification used, the procedure is inefficient; there is no reason why all gaps behind a small aircraft should be stretched alike, for instance, even if they did all start out the same size. Obviously, trying all possible combinations of gap stretches for 16 different separations would be an overwhelming task; even if only five different values were used to more accurately characterize the entire DLTAIJ matrix, an already-expensive technique would be made much more costly.

Use of the SUPER subroutine for gap stretching is definitely not recommended. It was not removed from the upgraded capacity program, however, because it includes other functions as well, and can be used as a check on the gap stretching technique which was finally adopted.

2.2.2 Other Incremental Techniques

Two other simpler techniques have also been suggested for gap-stretching. One would involve a "floor value" for DLTAIJ which would be raised in increments, while the other would do the same

---

* At the time SUPER was written, three numbers were adequate to describe the separation standards then in effect (3/4/5 nmi). Even then, however, SUPER did not accurately condense the separation matrix -- LH should be the same as LL, not HH. The use of four separate separation values, begun in November 1975, is completely beyond the capabilities of SUPER.

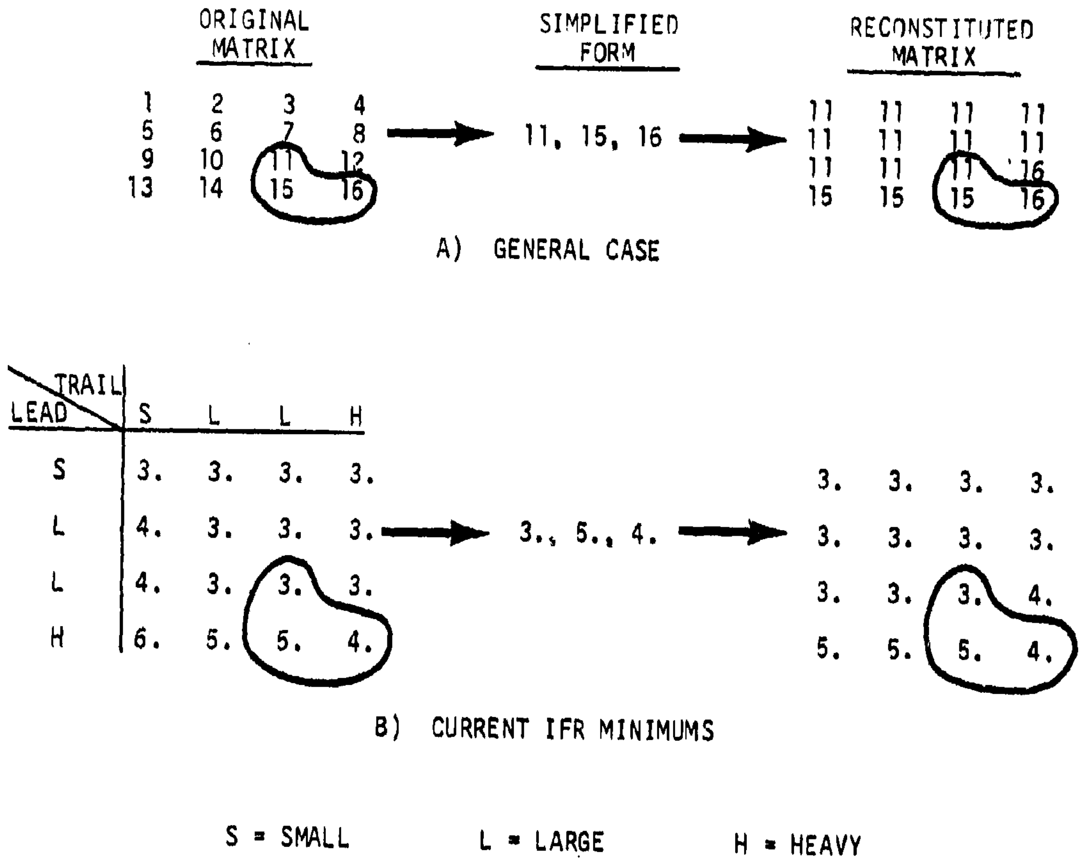


FIGURE 2-1
SIMPLIFICATION OF THE DLTAIJ MATRIX BY SUBROUTINE SUPER

for the interarrival time; the difference between the two sets of results is solely due to the velocity differential between aircraft. These techniques attempt to duplicate what is perceived to be the method by which controllers typically increase departure capacity: if separations are initially 3., 4., 5. and 6. nmi, first stretch them to a minimum of 4.0 nmi, then to a 5.0 nmi minimum, and finally to all 6.0 nmi spacings or greater.

Unfortunately, there is no guarantee that such a simple stretch will actually increase departure capacity, and it might in fact lead to less capacity, not more. For example, the natural gap between two arrivals might be just large enough for one departure, on the average. If the gap were stretched by an arbitrary amount, it might wind up being too small for two departures, but more than enough for one departure -- what we will refer to as an "inefficient" size. In this case, departure capacity would decrease, since although the number of departures per arrival gap remained the same, the number of arrival gaps per hour would be less at the larger spacing.

The curves of arrivals vs. departures which result from these techniques (Figure 2-2) reflect the move from efficient to inefficient back to efficient gap sizes. The curves are not smooth, but rise and fall like a roller coaster. Eventually the curve plunges towards the origin. The program contains an internal limitation of three departures per arrival gap, and after this limit is reached, any additional gap stretch would only reduce arrival and departure capacity, which remain in the ratio of one to three.

An alternative to these roller-coaster curves would be to construct a convex hull to each curve (that is, a series of straight-line segments drawn tangent to the outside of the curve), on the realistic premise that operation would only occur at efficient points on the curve. This would be fine, except for a basic weakness of this technique, shared with SUPER: gaps are stretched indiscriminately, without regard to whether or not the stretch is beneficial for an individual arrival pair. Overall capacity might increase, compared to the prior results, but not by as much as it possibly could if more selective gap stretching techniques were employed.

2.2.3 Selective Gap Stretching

Several techniques were investigated for stretching specific arrival gaps to a precalculated optimum size. These selective techniques differ from the previously discussed incremental techniques in several respects:

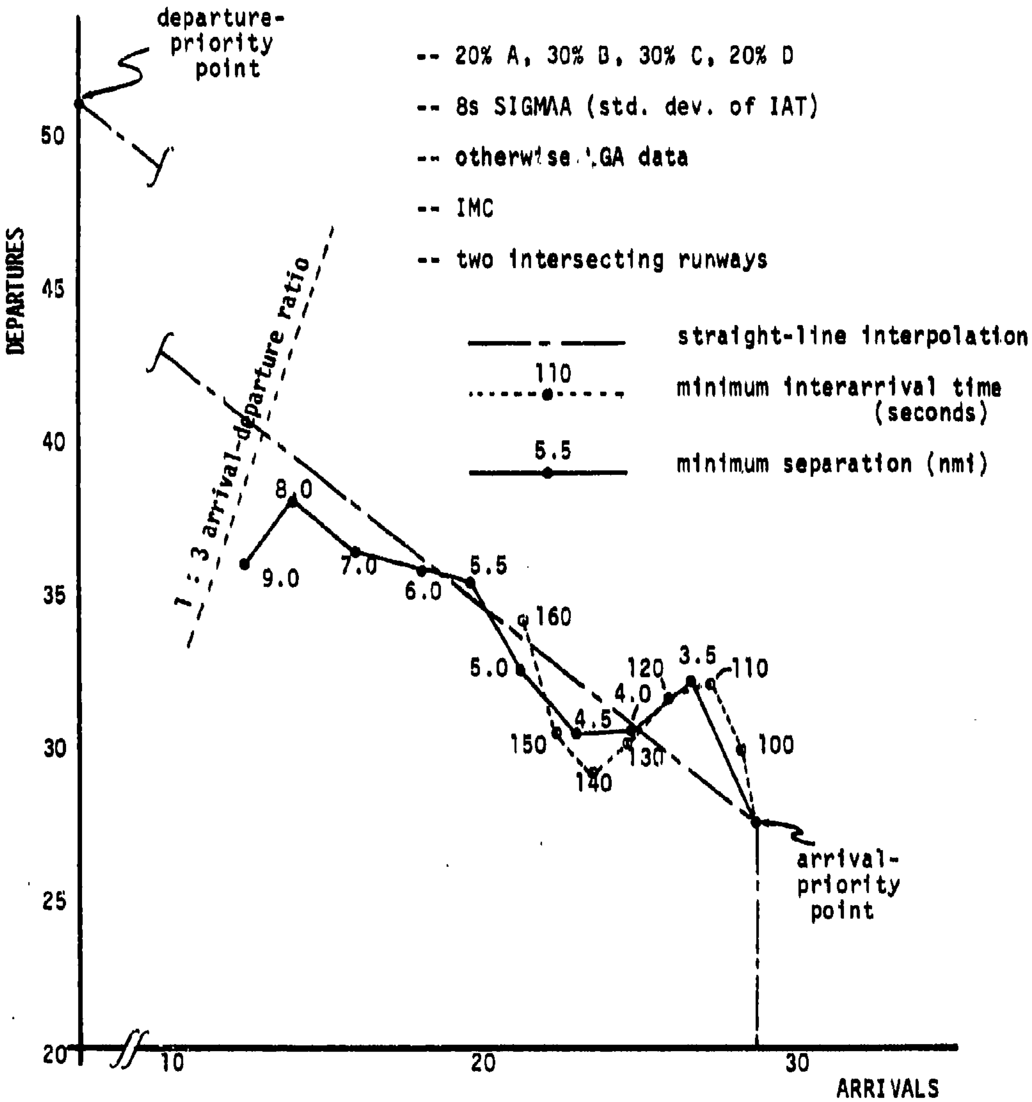


FIGURE 2-2
CAPACITY CURVE FOR SIMPLE INCREMENTED TECHNIQUES

- Gaps between individual arrivals are being stretched vs. gaps between several different aircraft types -- selective techniques are finer-grained

- Gaps are being stretched to a calculated optimum size vs. by a pre-determined increment -- selective techniques are less arbitrary

- Gaps can be stretched in the order of greatest capacity benefit first vs. strictly according to the incremental procedures used -- selective techniques are potentially more efficient.

The major drawbacks to selective gap stretching are cost and computational complexity. For these reasons a pure selective technique was not implemented in the revised program.

The first selective technique tested looked at all possible combinations of five aircraft (i,j,k,l, and m) and calculated the minimum size for the gap between arrivals i and j in order to allow one departure (k), two (l), or three (m). If the original gap were at least x% of the required size, it would be stretched to that size. The percentage x ranged from 100% downwards, to produce the full capacity curve.

This technique did not produce satisfactory results. The major failing was a sharp drop in capacity with larger amounts of gap stretching. It is believed that this occurred because the effect of the gap stretching on other departures was ignored. For example, stretching a gap to get out the second departure with a high probability would also increase the probability of releasing the third departure. Gaps were not being stretched to the most "efficient" size: the expected number of departures relative to the size of the gap was not being maximized.

The second technique tested was to attempt to stretch each gap to an optimum size for each k,l,m combination. The gaps were then ordered so that the stretches which provided the maximum benefit were performed first. Also, a gap was not stretched if greater benefit could be obtained by running departures-only at the end of the hour instead.

This technique was more successful, but still not completely satisfactory. The downturn in the capacity curve was avoided, and higher capacity values were obtained (Figure 2-3). However, the calculations were expensive because of the large number of combinations being studied and the iterative method used to find the

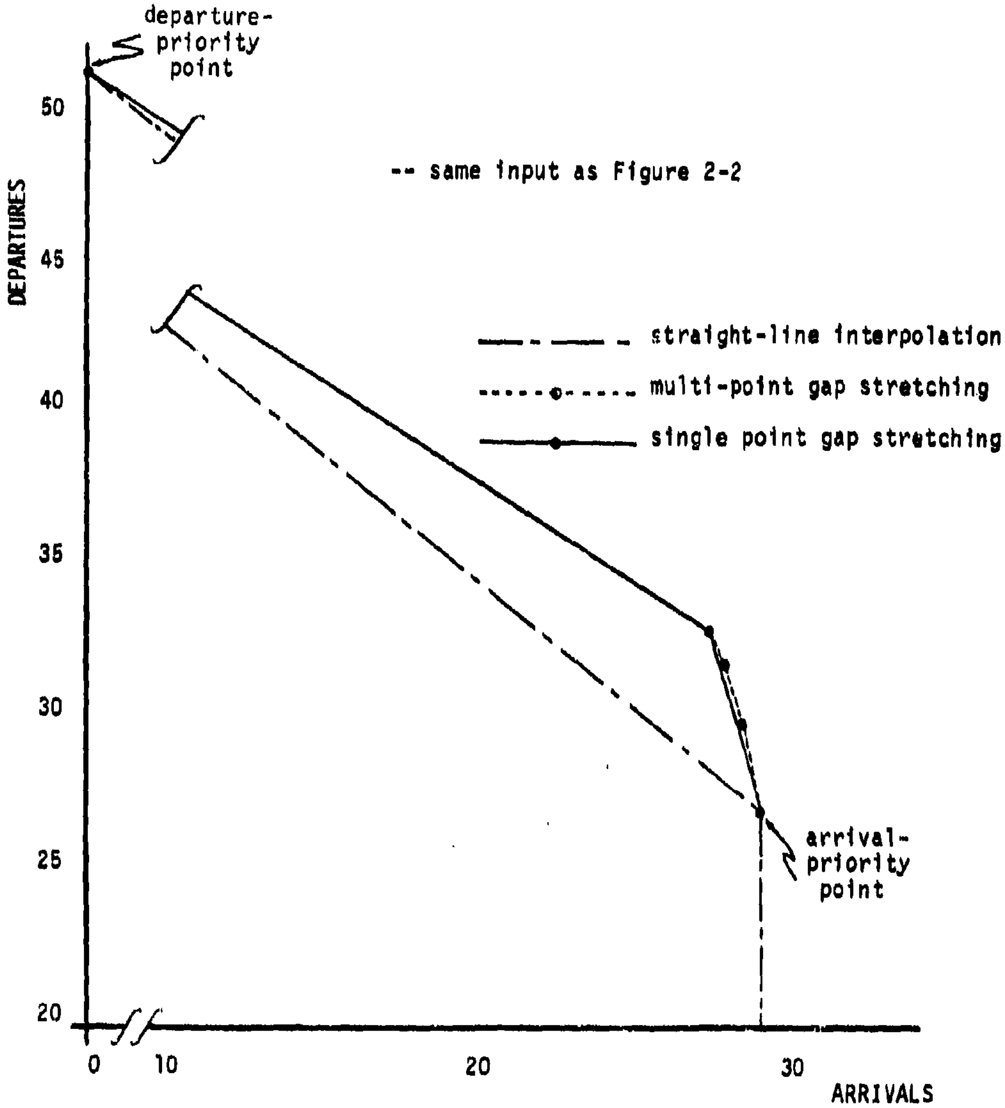


FIGURE 2-3
CAPACITY CURVES FOR SELECTIVE GAP STRETCHING TECHNIQUES

optimum gap size. Also, this technique only dealt with local optimum points: it would stop searching as soon as any additional stretch would reduce the efficiency of the gap. It can and did occur that a greater stretch would result in a more efficient gap than the first local optimum which was obtained. Partially for this reason, this technique did not result in the largest capacity results obtainable.

Part of the cost of this technique came from ranking and ordering the various gap stretches. The logic was modified so that all beneficial gap stretches would be performed at the same time. This resulted in only one intermediate capacity point, at considerably less expense, but with the same shortcomings regarding local optima and capacity values.

More complete descriptions of these three selective techniques may be found in Appendix B.

Some fundamental questions about this form of selective gap stretching are yet to be resolved. They relate to the concept of stretching gaps to particular sizes to fit particular departure sequences:

- Is it valid to consider stretching gaps to fit particular departure sequences, when in reality the arrival spacing is set up too far in advance for the controller to possibly know the departure sequence?

- What happens when the last departure in the sequence cannot fit into the current gap, and the next gap is tailored to a different sequence of aircraft? Stretching a gap to an optimal size with respect to the expected value of departures within the gap results in a non-negligible probability of missing the last departure.

The answer to these questions might be that we are interested in the <u>maximum</u> throughput, and the situations described are no less realistic than the assumption of infinite supplies of arrivals and departures. But the argument is made moot by the other shortcomings of the selective technique, which led to eventual abandonment.

2.3 Selective Incremental Interarrival Time

The gap stretching technique which was finally chosen and implemented in the revised capacity program combined features of both the incremental and the selective techniques described. The basic procedure is incremental: a fixed increment of time is added to

each unstretched interarrival time (IAT). However, the decision of which gaps to stretch and by how many multiples of the basic increment is done on a selective basis. Incremental stretches are added temporarily to each gap but are retained only if the stretch makes that gap more efficient.

The following is a basic description of how selective incremental gap stretching works (refer to the flow chart, Figure 2-4):

- Unstretched gap sizes (interarrival times) are computed first (TAA(i,j)). There are sixteen values, based on four possible aircraft types for the lead arrival and four for the trail aircraft.

- For each gap, an expected departure rate is calculated and stored. This rate is the expected number of departures in that gap, considering all possible sequences of one to three departure types and their probability of occurrence in the fleet, divided by the size of the gap in seconds.

- Hourly arrival and departure capacities are computed, in the usual manner, for this unstretched case.

- The user-specified increment (DELTA) is then added to each interarrival gap, and the expected departure rate for each gap is recomputed using the new gap sizes (TAATST(i,j)).

- The gap remains at the new size if the new departure rate is higher than for the alternative, which would be to operate with an unstretched gap and then to run departures-only mode during a time period at the end of the hour equivalent to the increment. If there is no capacity benefit from the stretch, the gap is restored to its previous size.

- Arrival and departure capacity are recalculated, given the resulting matrix of stretched and unstretched interarrival times. This gives the first intermediate point on the capacity curve.

- To get the next intermediate point, all interarrival times are set to their unstretched size plus twice the pre-set increment, and this test value is used to calculate the new expected departure rate. After the comparison of expected departure rates, some interarrival times might remain with the double stretch,

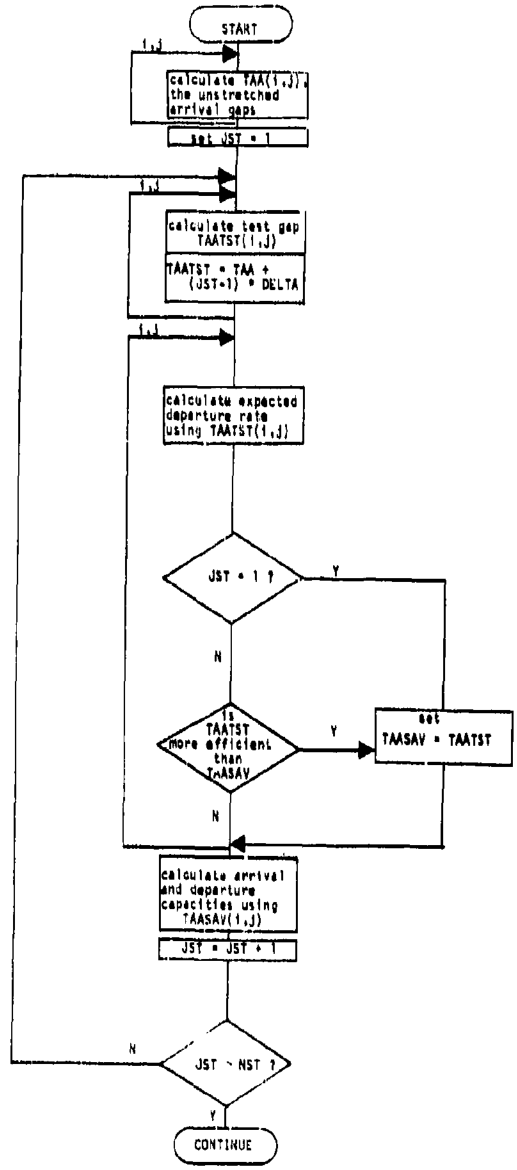


FIGURE 2-4
FLOW CHART FOR SELECTIVE INCREMENTAL INTERARRIVAL TIME TECHNIQUES

while the others revert to their previous size of single stretch or no stretch.

- For all successive capacity points, up to the limit specified by the user (NST), the same pattern is repeated: all interarrival gaps are stretched by the same amount, but the increment is retained only in those cases where a capacity benefit results.

This technique was chosen for several reasons: it was philosophically acceptable (as opposed to selective techniques based on prior knowledge of the departure queue), and it produced larger capacity increments at lesser cost of running time than did any of the other techniques tried. Table 2-1 presents a comparison of the different techniques in terms of running time and capacity, for one example.

Other comparison cases were run as well in the course of testing. Naturally the relative costs and capacity increases shown in Table 2-1 did not always hold true.* However, based upon the cases which were run and the characteristics of the techniques which were tested, the method chosen was the best available.

The program user can specify both the number of capacity points to be calculated and the IAT increment to be used. If these values are not specified, the gap stretching logic and the interactive program version default to a calculation of one additional point at an IAT increment of 20 seconds. The user has the option of calculating any number of capacity points from one (arrival-priority) to twenty (arrival-priority plus nineteen intermediate points).

In some cases where gap stretching would not be appropriate, such as an arrivals-only runway, it does not occur. The gap stretching logic is incorporated in only three simple configurations -- single runway with mixed operations (1-3(B), subroutine MIXOP),

* The program running costs quoted in this report were obtained during program development. After development was complete, the revised version was placed in the more efficient MODULE form, which is faster for MITRE's IBM system to load than the series of separate TEXT files, one for each subroutine, which comprised the program during development. The result is a typical savings of 2 CPU seconds per run, which should be considered in any comparison between the revised version and the original version. The original version is already a MODULE.

TABLE 2-1

COST AND CAPACITY COMPARISON OF ALTERNATIVE CAPACITY TECHNIQUES

	ORIGINAL VERSION	INCREMENTAL MINIMUM SEPARATION	MULTI-POINT GAP STRETCHING	SINGLE POINT GAP STRETCHING	SELECTIVE INCREMENTAL INTER-ARRIVAL TIME
NUMBER OF INTERMEDIATE POINTS	1	5	5	2	2
MAXIMUM CAPACITY	56.1 (52% ARR)	59.0 (45% ARR)	60.25 (45% ARR)	60.25 (45% ARR)	58.5 (45% ARR)
50% ARRIVAL CAPACITY	55.9	56.9	57.2	57.1	56.8
COST (CPU seconds)	10.9	31.4	92.7	29.6	11.4

-- IMC, TWO INTERSECTING RUNWAYS

-- MIX 20%A, 30%B, 30%C, 20%D

-- 8s STANDARD DEVIATION OF IAT (SIGMAA)

-- OTHERWISE LGA DATA

close parallel arrivals and departures (2-20(C:A,D), subroutine TWOPA) in IMC, and intersecting arrivals and departures (6-2(A,D), subroutine TWOIN) -- and will only be employed if one of these configurations, or another configuration which shares the same logic, is being analyzed.

The same configurations which utilize gap stretching are the ones for which the "equal-priority" option was previously available. Under this option, a 50% arrival-departure mix was enforced by stretching each arrival gap, if required, to allow one and only one departure in each gap. As a gap-stretching technique, this was crude and also fairly arbitrary: if the gap was already large enough for a single departure it would not be stretched further, no matter how inefficient it was; also, the gaps which were stretched were not necessarily stretched to an optimum size.

The equal-priority logic was only intended originally to be a special-case alternative to linear interpolation, not a complete solution. Selective gap-stretching is the preferable technique in all circumstances. Consequently, the equal-priority option is no longer available.

3. OTHER MAJOR LOGIC CHANGES

3.1 First Enqueued Departure Mix

3.1.1 Departure Capacity and the Mix Constraint

The FAA Airfield Capacity Model assumes that the fleet mix of the departures -- the relative proportions of the four different aircraft types -- will be identical to the mix of arrivals on the same runway. This is certainly a reasonable assumption, especially since the capacity program is based upon long-term average operating characteristics. Over any long term (usually a day or more), what goes into an airport must come out again.

The need to satisfy a given departure mix introduces some complexity and inefficiency into the calculation of departure capacity. This section will explain the means by which the original program version dealt with the departure mix constraint, the shortcomings of that technique, and the approach taken by the revised version.

First, an explanation of the departure capacity calculation is required. The departure capacity of a mixed arrival-departure runway is based on the premise that arrivals have priority, and departures must be inserted as possible into the resulting interarrival gaps. P1(ijk) is defined as the probability that one or more departures can be released in the gap between arrivals i and j, given that the first departure is of type k. In other words, P1(ijk) is the probability that the ij gap will be as large or larger than required to release departure k.

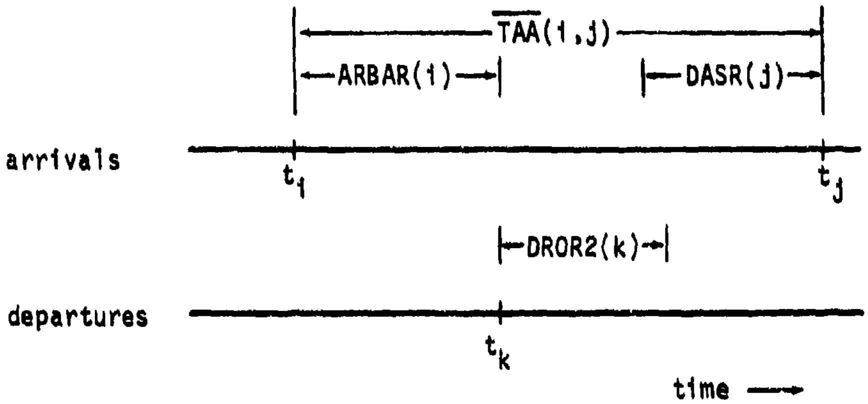
The size of the interarrival gap is assumed to be normally distributed, with mean $\overline{\text{TAA}}(ij)$ and standard deviation SIGMAA. The expression for P1 is therefore (see Figure 3-1):

$$\text{P1}(ijk) = \phi\left[\frac{\overline{\text{TAA}}(ij) - \text{ARBAR}(i) - \max\,(\text{DASR}(j),\ \text{DROR2}(k))}{\sqrt{\text{SIGMAA}^2 + \text{SIGMAR}^2}}\right] \quad (1)$$

where ARBAR(i) = average runway occupancy time of arrival i

DASR(j) = departure-arrival separation requirement, the time for arrival j to fly DLTADA

DROR2(k) = the protected departure occupancy time of k



where t_i = the time at which arrival i crosses the threshold

t_j = the time at which arrival j crosses the threshold

t_k = the time at which departure k is released

FIGURE 3-1
TIME AXIS DIAGRAM OF SINGLE RUNWAY OPERATIONS

SIGMAR = the standard deviation of the arrival occupancy time

Φ = the cumulative distribution function operation.

This expression applies standard probabilistic concepts to evaluate the probability that the ij gap is more than large enough for a single departure.

Similar equations are used for P2(ijk,l) and P3(ijk,l,m), the probabilities of two or more departures and three departures, respectively (the probability of more than three departures per gap is assumed to be zero).

Given these probabilities, it is a simple matter to calculate the expected number of departures of a particular type per gap. Calling this value D(k),

$$
\begin{aligned}
D(k) = \; & P\,[\text{one or more departures, the first one of type } k] \\
& + P\,[\text{two or more departures, the second one of type } k] \\
& + P\,[\text{three departures, the third of type } k] \\
= \; & \sum_{ijxy} [P1(ijk) + P2(ijx,k) + P3(ijx,y,k)] * \\
& \%i * \%j * \%x * \%y * \%k
\end{aligned}
\qquad (2)
$$

where %i = the proportion of type i in the mix.

Due to differences in occupancy times, departure-arrival and departure-departure separation requirements, some aircraft types may be more likely to be released in an arrival gap than others. Unless this was compensated for, the departure mix would be distorted in favor of the more easily released types.

The original program avoided mix distortion by basing departure capacity on the most <u>limiting</u> aircraft type, defined as the type with the smallest value of D(k)/%k. Departure capacity is then set equal to

$$\text{departure capacity} = \text{number of arrival gaps} * \min\left[\frac{D(k)}{\%k}\right]. \qquad (3)$$

However, this procedure does not consider the operational characteristics of a runway system which tend to minimize the impact of the limiting aircraft type. For example, suppose that only one in ten arrival gaps was large enough for a departure of type B, which comprised half of the mix. The joint probability of a large enough arrival gap and a departure of type B would be 0.1 X 0.5, or 0.05; this would be the expected number of departures per gap of type B, or D(B). Departure capacity would then be 0.05/0.5, or 10% of the arrival capacity.

In reality, however, departure capacity would be greater. A departure of type B might have a 50% chance of being the next departure in the queue, but once it reached the runway, it would wait there until a large enough arrival gap appeared. On the average this would require ten arrival gaps. If a type A departure required only one arrival gap, and a type B required ten on the average, the departure rate would be two departures in eleven arrival gaps, or 18.2% of the arrival capacity -- an 82% improvement over the original logic.

This example presents an extreme case unlikely to be seen in reality but designed to make a point: the existing departure capacity logic can underestimate true capacity in order to avoid distorting the departure mix. Basing capacity on the most limiting aircraft type is a reasonable approach to the problem; what is questionable is the departure rate being calculated for that type.

3.1.2 Concept of the f.e.d. Mix

The departure rate for a particular aircraft type is a function of both the probability that the type can be released in an arrival gap and the probability that the type will be in position to depart in that gap. Since our definition of capacity as maximum throughput requires that there always be a queue of aircraft waiting to depart, this latter probability can only relate to the type of aircraft which is first in line to depart.

The original capacity logic assumes that the probability of a particular type being first in the queue is equal to the overall proportion of that type in the mix. In reality, the first aircraft in the queue to depart in a given gap will be the aircraft which was unable to depart in the previous gap. The first enqueued departure (f.e.d.) mix will thus be distorted to favor the most limiting aircraft type. This concept of a separate f.e.d. mix for departures recognizes that once a departure reaches the runway, it waits there until a suitable gap is available; the

original logic said, in effect, that such an aircraft would return to the end of the queue if it could not depart in the first available gap.

Unfortunately, there is no closed-form solution to obtaining the f.e.d. mix. The probability that a particular type will be first in the queue is the probablity that it cannot be released in the previous gap, which in turn depends upon the f.e.d. mix for that gap. An iterative method is required, in which the f.e.d. mix is calculated based upon the mix calculated during the previous interation. Fortunately the mix converges rapidly. Figure 3-2 presents a flow chart of the f.e.d. mix logic. The f.e.d. mix is described by the variable PFED(i,k), which is defined to be the probability that the first departure in the queue after an arrival of type i will be type k. Specification of the arrival i is important. The probability that k is first in gap ij actually depends on k not fitting into gap gi, but looping over the type of the previous arrival g as well (i.e., PFED(g,i,k)) would cause additional running cost and fortunately is not necessary.

For the first iteration through the f.e.d. mix computation, the temporary variable PTEMP(i,k) is set equal to the overall fleet mix of type k (PHR(k)). The departure probabilities P1, P2 and P3 are calculated in the usual way, but using PTEMP(i,k) as the probability that the first departure is type k. The f.e.d. mix is then calculated by evaluating the probability that k will not be released in the gi gap.

$$\begin{aligned} \mathrm{PFED}(i,k) = & \sum_{g} [1.0 - \mathrm{P1}(gik)] * \%g * \mathrm{PTEMP}(g,k) \\ & + \sum_{gx} [\mathrm{P1}(gix) - \mathrm{P2}(gix,k)] * \%g * \mathrm{PTEMP}(g,x) * \%k \\ & + \sum_{gxy} [\mathrm{P2}(gix,y) - \mathrm{P3}(gix,y,k)] * \%g * \mathrm{PTEMP}(g,x) \\ & \qquad * \%k * \%y \\ & + \sum_{gxyz} \mathrm{P3}(gix,y,z) * \%g * \mathrm{PTEMP}(g,x) * \%k * \%y * \%z. \end{aligned} \tag{4}$$

The first term evaluates the proportion of time when k is the first departure in the queue, but the gi gap is not large enough. The second and third terms refer to k as the second and third departures, respectively. The last term considers the possibility that the maximum of three departures are released in the gi gap, and k is the next departure in the queue.

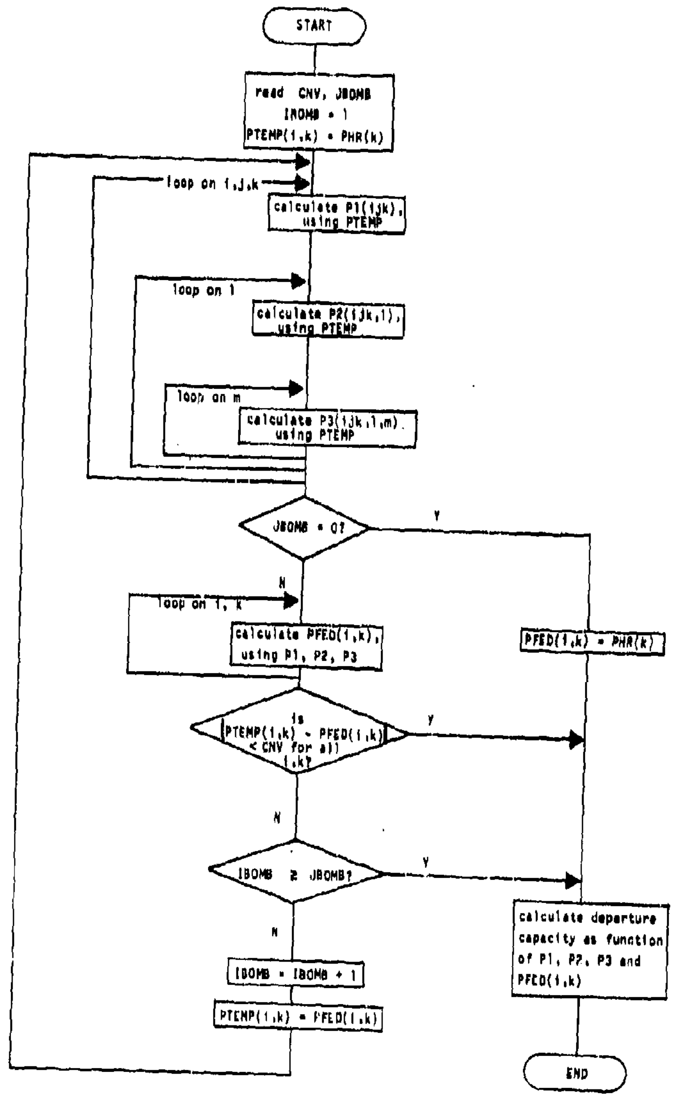


FIGURE 3-2
FLOW CHART FOR f.e.d. MIX LOGIC

Finally, a check for convergence is made. If the difference between PTEMP(i,k) and PFED(i,k) is less than the user-input convergence criterion (CNV), or if the specified maximum number of iterations (JBOMB) have been performed, the iterations stop and the last values calculated for P1, P2, P3 and PFED(i,k) are used to determine departure capacity. The convergence criterion is currently an algebraic difference between successive values, but a proportional difference could be used with only minor program changes.

A useful estimate of the f.e.d. mix will be obtained on the first pass through the program logic. If the f.e.d. mix is not desired, specifying zero as the maximum number of iterations (JBOMB) will cause the overall fleet mix to be used instead for departure capacity calculations, as is done in the original program.

It should be noted that the f.e.d. mix can affect the gap stretching logic discussed in Chapter 2 and vice versa. Stretching all the gaps will definitely change the probabilities of one, two, or three departures (P1, P2 and P3) and therefore the first enqueued departure mix, but changing the f.e.d. mix can also change the choice of which gaps to stretch. And, of course, restoring some of the gaps to their previous size will affect the f.e.d. mix again.

Ideally, the f.e.d. mix would be calculated both before and after the stretched gaps are selectively unstretched. This was not done due to the extra cost that would result. The f.e.d. mix is calculated only before the gaps are restored. The result seems to be a slight, but acceptable, underestimate of capacity, based upon test cases which were run.

3.1.3 Implementation

In a previous section we presented an extreme example to demonstrate the need for the f.e.d. logic. When this case was run through the revised capacity program, the f.e.d. logic produced the answer which had been pre-calculated: departure capacity was 18.2% of arrival capacity.

Besides using some unrealistic values in order to make a point, this case was also unrealistically simplified to allow quick hand calculation of the answer. Only two aircraft classes with identical arrival characteristics and operational characteristics were chosen so that no more than one departure per gap could ever occur -- features which would never be seen in reality.

Subsequently, some more realistic cases were run to evaluate the effect of the f.e.d. logic on cost and running time. Miami International Airport was used as the test case; input data was obtained from the Airport Capacity Task Force final report, Reference 5. The case was run three times: without the f.e.d. logic, with only one iteration through the f.e.d. logic, and with several iterations. Results are shown in Table 3-1.

Table 3-1 also presents cost data for the three runs in terms of total CPU seconds. The running times for certain sections of the program are shown (obtained by calling a special timing function of the computer system), as well as the total time which is returned by the operating system when the program is completed.

'READ' indicates the time required to read the input file until the calculations begin. 'CALCULATE' includes the time to calculate the arrival priority and departure priority capacities, as well as any intermediate points, and compute the capacity at the desired percentage of arrivals. The total includes these times and also that required to load the program and print the results.

Specifying one iteration through the f.e.d. logic increased running cost by 9% (14.25 CPU seconds vs. 13.09) but increased the arrival priority departure capacity by 7% (from 16.71 to 17.92) and the capacity at 50% arrivals by 3.5% (from 54.2 to 56.1). Additional iterations are much more expensive because P1, P2 and P3 must be recalculated (see flow chart, Figure 3-2). When a maximum of five iterations was specified, running time went up 178%, but the largest capacity increase was 15%.

Some variations can be expected in the cost differential between f.e.d. runs and non-f.e.d. runs for other cases, of course. If the number of iterations is determined by the specified convergence criterion, a case which does not converge as quickly would entail more iterations and higher expense. Also, because the total CPUs required for a job depends upon the load on the time-sharing system, the same run might not cost the same two days in a row. The READ should cost the same in each of the three MIA runs but it does not, for these reasons.

This understanding of these CPU time numbers should prevent us from relying too heavily on the actual values in Table 3-1. However, the qualitative message is unchanged: the f.e.d. logic can be useful in small doses but can be expensive if not used carefully.

TABLE 3-1

EFFECT OF f.e.d. LOGIC -- VMC CASE

	NO f.e.d. MIX		1 f.e.d. ITERATION		5 f.e.d. ITERATIONS	
CAPACITY	ARR	DEP	ARR	DEP	ARR	DEP
A.* ARRIVAL PRIORITY	36.59	16.71	36.59	17.92 (+7%)	36.59	19.20 (+15%)
DEPARTURE PRIORITY	0.0	49.91	0.0	49.91	0.0	49.91
20s STRETCH	30.84	24.00	30.84	25.90 (+8%)	30.72	26.64 (+11%)
50% ARR	54.2		56.1 (+3.5%)		56.8 (+4.8%)	
B.* ARRIVAL PRIORITY	36.59	16.59	36.59	17.76 (+7%)	36.59	18.95 (+14%)
DEPARTURE PRIORITY	0.0	48.64	0.0	48.64	0.0	48.64
20s STRETCH	30.84	23.84	30.84	25.61 (+7%)	30.84	26.22 (+10%)
50% ARR	53.9		55.7 (+3.3%)		56.3 (+4.5%)	
COST (CPU seconds)						
A. READ	0.50		0.51		0.45	
CALCULATE	4.63		5.12		16.53	
B. READ	0.09		0.04		0.06	
CALCULATE	4.62		5.03		16.16	
TOTAL	13.09		14.25 (+9%)		36.42 (+178%)	

-- MIA TODAY, VMC INPUTS

-- SINGLE RUNWAY, MIXED OPERATIONS

*A. VFR DEPARTURE-DEPARTURE SEPARATIONS

*B. IFR DEPARTURE-DEPARTURE SEPARATIONS

Table 3-2 presents another case in which the f.e.d. logic was used. The input data for this case was drawn from the Task Force report for John F. Kennedy International Airport (Reference 6); IFR capacity is being calculated. In this case, the f.e.d. logic produced only a 0.2% increase in the arrival priority departure capacity and no change in the 50% arrival capacity. However, the extra running cost was only 5%.

In the JFK case, there was much less difference between the departure types than for MIA. The difference between the largest and the smallest departure runway occupancy time was 5 seconds (39 vs. 34) as opposed to 13 seconds (42 vs. 29); the JFK case also used non-standard IFR departure-departure separations (roughly 75/90/120) as opposed to the MIA VFR separations (35/90/120).

When differences in departure characteristics are substantial, the f.e.d. logic shows significant benefits. But even though the benefit may not be significant, the additional cost of obtaining the first f.e.d. mix is small. Consequently, the program has been written to default to using the first f.e.d. mix, calculated during the first pass through the logic (i.e., JBOMB = 1).

If additional accuracy is desired, we would recommend that the user rely upon the convergence criterion CNV rather than the maximum number of iterations JBOMB to limit the f.e.d. calculations. A reasonable value of CNV (such as 0.01) almost always limited the number of iterations to two or three in the test cases which have been run. The f.e.d. mix appears to converge rapidly and has never been observed to diverge. The number of iterations required to achieve a given level of accuracy has been related to the size of the error which would result by not using the f.e.d. mix. When the convergence criterion is used, therefore, multiple iterations are performed only when they are needed. For example, the VMC case above required three iterations to converge to 0.01, while the IMC case only went through one iteration. When running expense is a consideration, economy measures such as the use of multiple arrival percentages (Section 4.1) can be taken to hold down costs.

The f.e.d. logic has been added to subroutines MIXOP (single runway), TWOPA (two parallels), and TWOIN (two intersecting runways).

TABLE 3-2

EFFECT OF f.e.d. LOGIC - IMC CASE

	0 ITERATIONS		1 ITERATION		
CAPACITY	ARR	DEP	ARR	DEP	
ARRIVAL PRIORITY	26.41	16.71	26.41	16.74	(+0.2%)
DEPARTURE PRIORITY	0	38.60	0	38.60	
20s STRETCH	23.72	21.25	23.72	21.22	(-0.1%)
50% ARRIVALS	44.6		44.6		(+0%)
COST (CPU seconds)					
READ	0.48		0.41		
CALCULATE	4.68		5.34		
TOTAL	8.57		8.97		(+4.7%)

-- JFK TODAY, IMC INPUTS

-- SINGLE RUNWAY, MIXED OPERATIONS

3.2 Convolved Probabilities and Other Improvements to Q-logic

3.2.1 Description of Problem

When the program calculates the departure capacity of a mixed mode (arrival and departure) runwav, the computation involved is relatively straightforward. The average time available for departures -- the time from first arrival off the runway to next arrival over the threshold -- is calculated first, and this is then compared to the time required for one departure, two departures, and three departures in order to derive a probable number of departures in that interarrival gap.

If the arrivals and departures are on separate runways, either dual lane or intersecting, the computation is much more complex. The time required for one, two, or three departures can now be affected by the separation required behind the last departure in the previous arrival gap. This was not a significant factor for mixed operations on a single runway; the departure could not be released soon enough for the departure-departure separation to be a factor, in most cases, because the previous arrival had to clear the runway first.

The logic for considering the departure-departure separations between gaps, which we shall refer to as the Q-logic, includes calculations of the probability that the last departure in the previous gap is a particular type, the interval between that departure and the first arrival, and the effect of that departure on the current gap. The Q-logic will be more fully explained in Section 3.2.2.

Two problems in the original model were traced to the Q-logic. The first we referred to as the "mix bug": if all characteristics (speed, runway occupancy, separations, etc.) were identical for the four aircraft classes, varying the mix of classes should not have affected capacity, but it did. This turned out to be a relatively straightforward coding error.

The second problem was more difficult to fix. It manifested itself as follows: departure capacity would be higher when the weather was just barely IFR (e.g., 900' ceiling) than when it was VFR (1000' ceiling), all else being equal. Phrased another way, capacity got better as the weather got worse. This turned out to be a direct result of the Q-logic in the original program.

3.2.2 Description of Q-logic

To understand the solutions to these problems, we must first explain the Q-logic. Reference to Figure 3-3, a time-axis diagram of the arrival-departure operations, should be helpful.

For each arrival gap between arrivals of type i and type j, the following probabilities are first calculated in the usual manner:

- P1(ijk) -- The probability of one or more departures in the ij gap, given that the first departure is type k.
- P2(ijk,l) -- The probability of two or more departures, given that the first two departures are k and l.
- P3(ijk,l,m) -- The probability of three departures of type k, l, and m.

Note that the probability of more than three departures per gap is assumed to be zero.

The next step is to calculate Q(n,i), the probability that the last departure before i is of type n. This can be evaluated knowing P1, P2 and P3 by looking at all the possible ways for n to be the last departure in the previous gap:

$$Q(n,i) = \sum_{gxy} [P1(gin) - P2(gin,x) + P2(gix,n) - P3(gix,n,y) + P3(gix,y,n)] * \%g * \%x * \%y * \%n \qquad (5)$$

where: %g is the proportion in the mix of the previous arrival g

%x, %y are the proportion of the other departures x and y.

Next the original program calculated π, the probability that the departures in the current gap are not affected by previous departure n. It is assumed that the actual time at which n is released (termed t_n) is uniformly distributed (a fair assumption, given that we know nothing about the departures and arrivals which precede n).

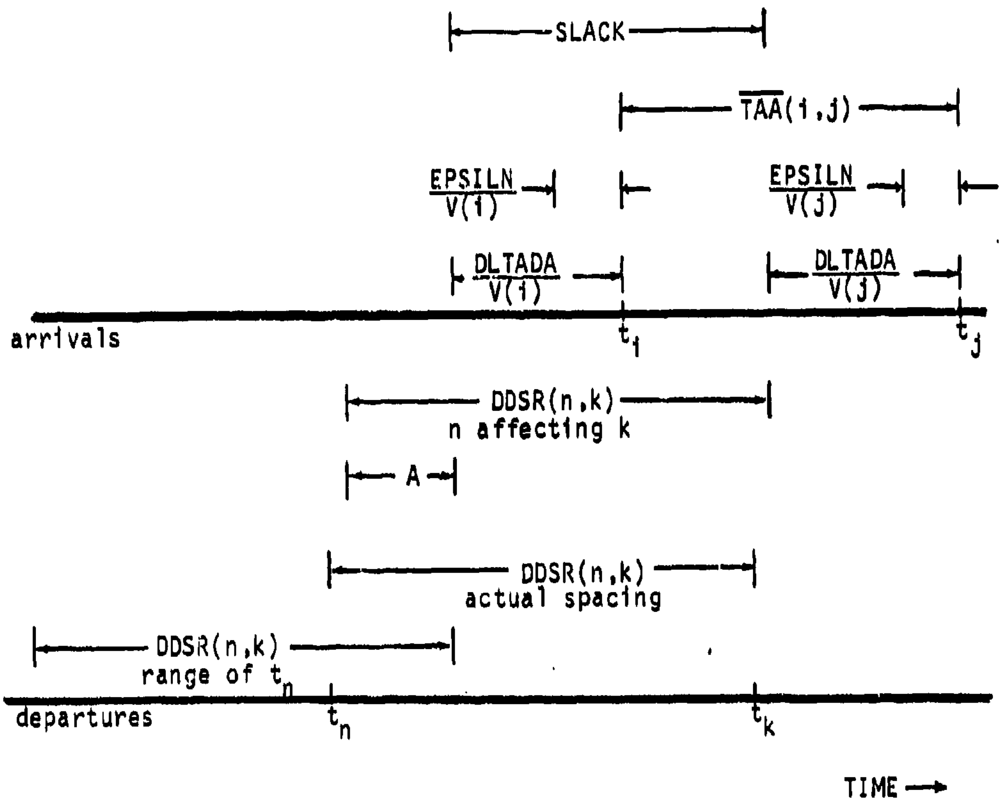


FIGURE 3-3
TIME AXIS DIAGRAM OF DUAL-LANE OPERATIONS

Departure n is bounded by DLTADA, the departure-arrival separation requirement -- n cannot be released if arrival i is closer than DLTADA to the threshold.

The possible range of release times for departure n is DDSR(n,k), the departure-departure separation requirement between departures n and k, because if n were to be released any sooner, it would no longer be the last departure in the previous gap -- k would be.

$\pi 1$ was then calculated.

$$
\begin{aligned}
\pi 1 &= P\ [\text{current departure } k \text{ is not affected by } n] \\
&= P\ [t_n \text{ is not within range } A] \\
&= 1 - \frac{A}{\text{range of } t_n} \quad \text{since } n \text{ is uniformly distributed} \\
&= 1 - \frac{DDSR(n,k) - SLACK}{DDSR(n,k)} \\
&= \frac{SLACK}{DDSR(n,k)} = \frac{\overline{TAA}(i,j) + \frac{DLTADA}{V(i)} - \frac{DLTADA}{V(j)}}{DDSR(n,k)}.
\end{aligned}
\tag{6}
$$

The probability $\pi 1$ assumes that there is a departure n in the previous gap. M1 is the probability over all n that the current departures are not affected.

$$
\begin{aligned}
M1 &= \sum_n \pi 1(n) * Q(n,i) \quad \text{-- the weighted sum of } \pi 1 \\
&\quad + 1 - \sum_n Q(n,i) \quad \text{-- the probability n does not exist in the previous gap} \\
&= \sum_n \%n * [1 - Q(n,i) + Q(n,i) * \pi 1(n)].
\end{aligned}
\tag{7}
$$

$\pi 2$, $\pi 3$, M2 and M3 are calculated similarly, except that the second and third departures l and m are considered.

Finally, the new probabilities P1*, P2* and P3* (probability of one or more, two or more, or three departures, respectively, considering the effect of any departures in the previous gap) were calculated. In the original logic,

$$P1^*(ijk) = (P1(ijk) - P2(ijk,l)) * M1 + P2(ijk,l) \quad (8)$$

= P [only one departure k and it is not affected, or two or more departures].

$$P2^*(ijk,l) = (P2(ijk,l) - P3(ijk,l,m))* M2 + P3(ijk,l,m). \quad (9)$$

$$P3^*(ijk,l,m) = P3(ijk,l,m) * M3. \quad (10)$$

3.2.3 Modifications to the Q-logic

The above section described the original form of the Q-logic for calculating the effect of departures in the previous gap on departures in the current gap. The logic is complex, but the basic concept can be summarized as follows: the actual probability of one or more departures (P1*), for example, is equal to the probability (P1) of one or more departures not considering the previous gap, times the probability (Q) that the last departure in the previous gap is type n, times the probability (π1) that departures in the current gap would not be affected by a previous departure of type n, summed over all values of n.

Such a summary is over-simplified, of course. Part of the complexity of the complete logic comes from including all values of n and the special case where departure n does not exist, and from the actual calculation of the probabilities. However, one simplifying assumption was made in the original program which is not valid: probabilities P1 and π1 are not independent, as the calculations would otherwise indicate.

Both P1 and π1 depend upon TAA(i,j), the time between arrivals i and j. The larger is TAA, the larger is P1, the chance of getting out a departure in that gap. Likewise, the larger TAA, the less chance that the current departure cannot be released because of the departure/departure separation required behind n, and therefore π1 is larger. The logic in the original program does not recognize this effect on π1, instead basing π1 on the average value of $\overline{TAA}(i,j)$. This concept of π1 is in error. If the ij gap size can, in effect, be "stretched" to accommodate one, two, or three departures, there may be no reason why it cannot be "stretched" a bit more to accommodate the required separation behind the previous departure n.

If both P1 and π1 depend on the value of TAA, a variable, then the two probabilities are not independent. P1* must be calculated as a single joint probability which considers both TAA as a

normally distributed variable and the release time of n as uniformly distributed. The convolution of the two distributions, resulting in new expressions for P1*, P2*, and P3*, is presented in Appendix C.

The new expression for P1*(ijk) cannot be explained in simple terms, due to the nature of the convolution. Some of the parts, however, may be familiar.

$$P1^*(ijk) = P1(ijk) + \sum_{n'} Q(n,i) \left\{ \left[\frac{\alpha}{DDSR(n,k)} - 1 \right] \left[PT1(n,ijk) + P1(ijk) - 1 \right] + \frac{SIGMAA}{DDSR(n,k)\sqrt{2\pi}} \left[E2 - E1 \right] \right\} \qquad (11)$$

where n' represents those values of n for which

$$DDSR(n,k) > (DLTADA - EPSILN)/V(i)$$

$$\alpha = \overline{TAA}(i,j) + DLTADA/V(i) - DLTADA/V(j)$$

$$PT1(n,ijk) = \Phi\left[\frac{DDSR(n,k) - \alpha}{SIGMAA}\right]$$

$$E2 = Exp\ [-\beta^2/2 * SIGMAA^2]$$

$$\beta = \overline{TAA}(i,j) + EPSILN/V(i) - DLTADA/V(j)$$

$$E1 = Exp\ [-(DDSR(n,k) - \alpha)^2/2 * SIGMAA^2].$$

The expressions for P2*(ijk,l) and P3*(ijk,l,m) are similar.

If the f.e.d. mix logic is utilized, subsequent iterations make use of the values of P1*, P2* and P3* to calculate new values of Q(n,i). The improved estimates of departure probabilities in the current gap also improve our estimates of departure probabilities in the previous gap.

3.2.3.1 Application to Parallel Runways

The original reason for this investigation into the Q-logic was the capacity results obtained for the dual-lane, close parallel runway case. The capacity results obtained after the revised expressions for P1*, P2* and P3* were implemented are contrasted with the original results in Figure 3-4.

The first thing which will be noticed from this illustration is that instead of a jump in capacity, there is now a slight drop in capacity at the point where the Q-logic is first employed. The drop is apparently caused by the limitation within the program of no more than three departures per arrival gap. The combination of large separation requirements behind heavy aircraft and increased spacing due to speed differentials can lead to interarrival times, for a small aircraft following a heavy, of four minutes or more -- enough in some cases for more than four departures. Any departures above three, however, are not being counted. This has been verified by running a test case without heavy aircraft in the mix, for which no capacity drop was evident.

If this is indeed the cause of the observed drop, then the impact of this limitation should decrease as ceiling and visibility tend towards zero. This is because the time for release of departures becomes shorter, and the chances of fitting in more than three departures per gap also shrink, as the value of EPSILN declines to zero.

The improvement in accuracy due to the use of the convolved logic does not necessarily increase the running time of the program. As Table 3-3 shows, costs of the original and revised versions are comparable, despite the increased complexity of the convolved logic. The principal reason for this is that P1*, etc., are calculated in one step; separate loops for calculating $\pi 1$, $\pi 2$, $\pi 3$ and M1, M2, M3 have been eliminated. Also, the calculations have been coded efficiently, so that items are not recalculated if their values haven't changed. Certain calculations are skipped entirely if they are not needed (i.e., if DDSR(n,k) < (DLTADA - EPSILN)/ V(i)), which explains why the cost of running the revised version went down as ceiling (and EPSILN) went to zero. Lastly, the calculations for $\Phi(x)$ were included in the subroutine, in order to avoid the overhead expense of calling subroutine PROB to do the calculation.

Implementation of the convolved logic also resolved the "mix bug" problem. This was due to a simple coding error which resulted in the double-weighting of M1, M2, M3 by the proportion of n in the fleet. The convolved logic eliminated a separate calculation of M, and therefore eliminated the "mix bug".

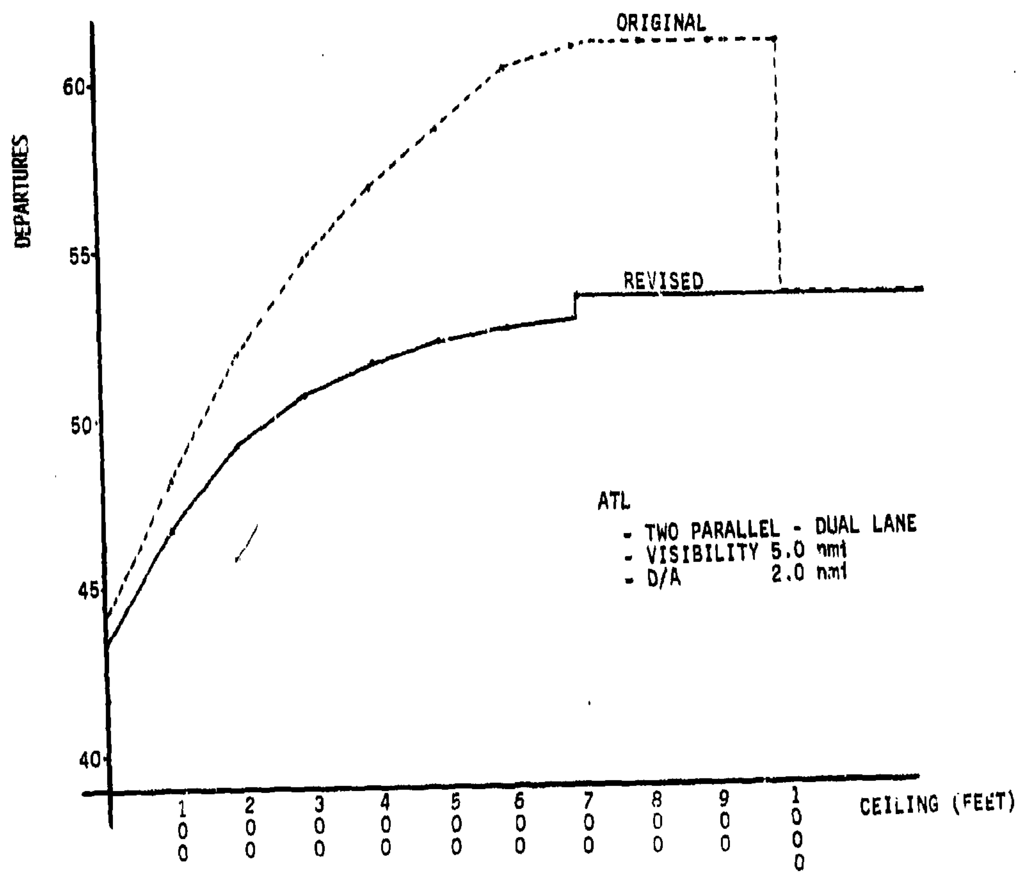


FIGURE 3-4
CEILING/VISIBILITY PROBLEM, DUAL-LANE RUNWAY

TABLE 3-3

COST COMPARISON OF CONVOLVED LOGIC

CEILING	ORIGINAL VERSION	REVISED VERSION
600'	11.02 CPUs	12.88 CPUs
0'	11.15 CPUs	10.86 CPUs

- ATLANTA DATA
- TWO PARALLELS - DUAL LANE
- VISIBILITY 5.0 mi
- DLTADA 2.0 nmi

The logic for calculating the capacity of a dual-lane runway in IMC has been isolated from subroutine TWOPA and placed in new subroutine DUAL. This was done for a number of reasons, but primarily to facilitate certain applications of the alternating arrival logic (see Section 4.2.3). It is hoped that the coding in DUAL can be generalized in the future for application to two intersecting runways and a single runway, so that the similar coding in subroutines TWOIN and MIXOP can be deleted.

3.2.3.2 Intersecting Runways

An intersecting pair of arrival and departure runways is operationally similar to a dual-lane pair in IMC: departures cannot be released after the arrival comes within a certain distance of the runway threshold (for the intersecting, to ensure that the departure clears the intersection in time; for the dual-lane, to comply with the departure/arrival separation requirement), until a certain time after the arrival crosses the threshold. The capacity calculation logic is also similar.

The convolved probability logic, however, has not been added to the calculation of P1*, etc. for intersecting runways. The main reason was that the ceiling/visibility problem could not exist for intersecting runways because the operation of intersecting runways does not depend upon EPSILN. For dual-lanes, some relief from the DLTADA restriction can be expected if the aircraft can see each other to apply visual separation; for intersecting runways, clearance at the intersection must be provided regardless of the visibility.

There are certain circumstances, such as runways with short intersection distances in VMC, where the convolved logic would improve the accuracy of the results obtained. Adding the convolved logic to subroutine TWOIN is therefore recommended for further investigation.

Other improvements to the Q-logic were made. The calculations for M1, M2 and M3 were changed to avoid the "mix bug". The correct equation for M1 is

$$M1 = \sum_{n} \%n - Q(n,i) + Q(n,i) * \pi 1(n) \tag{12}$$

which may be compared with the original equation (7) in Section 3.2.2 above.

The expressions for P1*, etc. were also modified slightly. The original expression for P1*,

$$P1^*(ijk) = (P1(ijk) - P2(ijk,l))^* M1 + P2(ijk,l)$$

stated that there would be one or more departures in the ij gap only if there were either just one departure in the gap which was not affected by n, or there were originally two or more departures per gap. A more complete expression for P1* would be

$$\begin{aligned} P1^*(ijk) &= (P1-P2) * M1 + (P2-P3) * M2 \quad + P3 * M3 \\ &\qquad + (P2-P3) * (M1-M2) + P3 * (M2-M3) \\ &\qquad\qquad + P3 * (M1-M2) \\ &= P1(ijk) * M1. \end{aligned} \tag{13}$$

In other words, it is sufficient to say that for one or more departures to remain in the gap, there must have been one or more departures to begin with, and the first departure k was not affected by the previous departure n. Similarly,

$$P2^*(ijk,l,m) = P2(ijk,l) * M2. \tag{14}$$

The expression for P3*,

$$P3^*(ijk,l,m) = P3(ijk,l,m) * M3,$$

remains unchanged. The impact of these changes was small but noticeable.

These and other modifications improved the accuracy and slightly lowered the cost of calculating the capacity of a 6-2(A,D) configuration. Costs and capacities are compared in Table 3-4.

3.2.3.3 Application of Q-logic to a Single Runway

The original version of the capacity program did not include the Q-logic in the capacity calculation for a single runway. There is usually no need to consider the departure-departure separations between gaps for the single runway because the separations are usually less than the combination of departure runway occupancy and arrival runway occupancy (which is not a factor for a dual-lane). However, departures in the current gap could be affected if previous departure n was a heavy, with a two-minute separation requirement. To test the impact of this, the Q-logic (without convolved probabilities) was added to the single runway subroutine.

TABLE 3-4

COMPARISON OF ORIGINAL AND REVISED INTERSECTING RUNWAY LOGIC

CASE	ORIGINAL VERSION		REVISED VERSION	
	CAPACITY*	COST	CAPACITY	COST
LGA mix	58.3	4.06 CPUs	57.4	3.77 CPUs
100 % type C	61.4	0.07	59.8	0.07
25% each type	48.1	3.88	46.6	3.65
TOTAL COST		8.63 CPUs		8.11 CPUs

- LGA DATA
- IMC (0/0) CONDITIONS

* CAPACITIES SHOWN ARE THE D1 VALUES -- DEPARTURE CAPACITY UNDER ARRIVAL PRIORITY

The impact of Q-logic is the greatest under conditions where the previous departure is likely to have the greatest effect, namely VMC, short arrival occupancy times, and a large percentage of heavies in the mix (and therefore large DDSRs). But as Table 3-5 shows, even under these conditions the difference due to the Q-logic was not large.

The decision to use Q-logic or not has been left up to the user. Normally, the program will bypass the Q-logic. To account for the departure-departure separations between gaps for a single runway, the user must input an artificial, negative value for DIAGSP, the first item on the ALTARR line of the input file.

A low priority was given in this project to single-runway Q-logic, in part because initial test runs using realistic data did not show a worthwhile effect. Consequently, the investigation to date is not sufficient to allow a decision as to whether or not to implement the Q-logic for all single runway cases. Some questions which remain to be answered include:

- Can definite guidelines be prepared for deciding on whether or not to use the Q-logic?

- Can the cost of the Q-logic be reduced (perhaps by only considering the cases where a heavy aircraft is the last previous departure)?

TABLE 3-5

EFFECT OF Q-LOGIC ON SINGLE RUNWAY CAPACITY

	Without Q-Logic		With Q-Logic	
VMC	arrivals	departures	arrivals	departures
arrival-priority	39.25	12.04	39.25	12.00
IAT + 10s	36.10	16.01	36.10	15.91
+ 20s	32.84	21.08	32.84	20.94
+ 30s	29.93	25.41	29.93	25.29
cost	10.1 CPUs		20.8 CPUs	
IMC				
arrival-priority	29.22	23.56	29.22	23.48
IAT + 10s	27.35	26.08	27.35	26.06
+ 20s	27.33	26.07	27.33	26.05
+ 30s	27.31	26.05	27.31	26.03
cost	9.4 CPUs		20.3 CPUs	

-- LGA data
-- 1%A, 13%B, 73%C, 13%D
-- current ATC system

4. NEW MODEL CAPABILITIES

The process of revising the original capacity model also included the addition of new capabilities which were not previously available. This section will discuss the following new features:

- Multiple arrival percentages
- Alternating arrivals
- Intersecting departure-departure runways.

4.1 Multiple Arrival Percentages

4.1.1 Usefulness

The capacity program calculates the maximum capacity under certain conditions (such as arrival-priority or departures-only) and then uses straight line interpolation to derive the capacity at a desired percentage of arrivals. In the original version of the program, only one arrival percentage could be specified per case. If the capacities at several arrival percentages were desired, several cases would have to be run, and the same arrival-priority and departure-priority capacity values would be calculated each time. In the revised version, up to 11 different arrival percentages can be specified during a single case, saving the expense of recalculating the arrival-priority and departure-priority capacities.

The capacities at several different arrival percentages are of interest for planning purposes to show how capacity (and delay) are affected by changes in the daily demand from arrival peak to departure peak. Capacities at 40%, 50% and 60% arrivals are typical of the desired outputs. Formerly, obtaining the three values would require that three separate cases be run, at a total cost of about 150% the cost of running a single case. With the revised version, the total cost for three, five or ten different percentages is only slightly greater than the cost for one.

4.1.2 Implementation

A flow chart for the multiple arrival percentage logic (in subroutine MAIN) is given as Figure 4-1. Simply stated, the logic calculates the capacity points, stores the results, and then interpolates between them. Some complexity is added by allowing for an unknown number of intermediate points (produced by selective gap stretching) which are calculated only as needed.

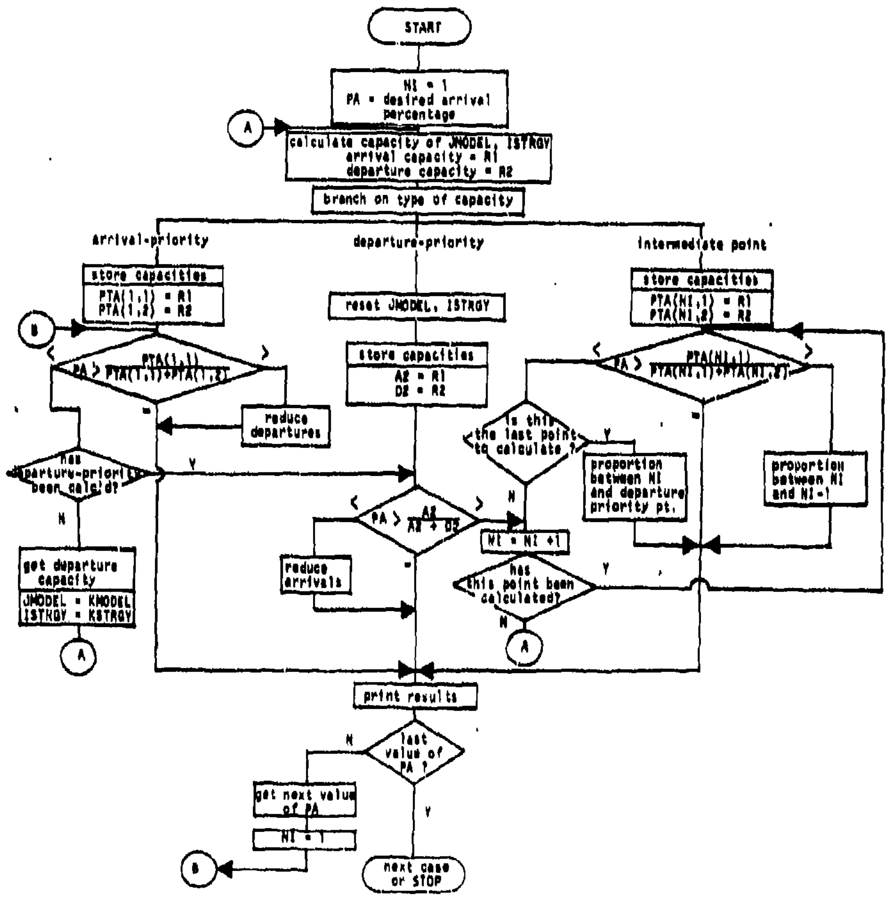


FIGURE 4-1
FLOW CHART FOR MULTIPLE ARRIVAL PERCENTAGE LOGIC

On the first pass through the logic none of the capacity points have been calculated except the arrival priority point (A in Figure 4-2). If the desired arrival percentage (PA) is greater than the percentage at this point, the excess departures are dropped. If less, the next step is to calculate the departure-priority capacity (point B). Since non-conflicting arrival streams are allowed under departure-priority, the arrival percentage at this point is not necessarily zero. If the desired arrival percentage is still less, the required number of non-conflicting arrivals is dropped. More likely, the desired point is between the arrival-priority and departure-priority points.

If only one point (i.e., arrival-priority) was specified, interpolation between A and B follows. Otherwise, the first intermediate point (C) is calculated. At this point the program checks for inflection -- has the capacity curve started to curve down towards the origin? The program checks whether this new point is below the line from the previous point to the departure-priority point. If so, this point is discarded and no new points are calculated. If not, the comparison with the desired percentage occurs again, and we either interpolate or calculate the next intermediate point. If the maximum specified number of intermediate points have been calculated, interpolation occurs between the last point and the departure-priority point.

No further intermediate points are calculated once a sufficient number have been obtained to calculate capacity at the desired percentage, even if the maximum number of points has not yet been attained.

Each of these capacity points has been stored in the array PTA(ni,j) as it was calculated. In the current program a maximum of 21 capacity points can be stored (the arrival priority point, 19 intermediate points, and the departure-priority point). The capacities at subsequent arrival percentages are computed using the values in PTA(ni,j) to the extent possible. There are no constraints on the order in which the desired arrival percentages can be specified, so for each new percentage the logic must start the search at the beginning. If the desired arrival percentage is greater than that at the departure-priority point, but less than that at the last calculated intermediate point, then additional intermediate points will be calculated up to the specified maximum.

The first arrival percentage specified can also take one of three special values:

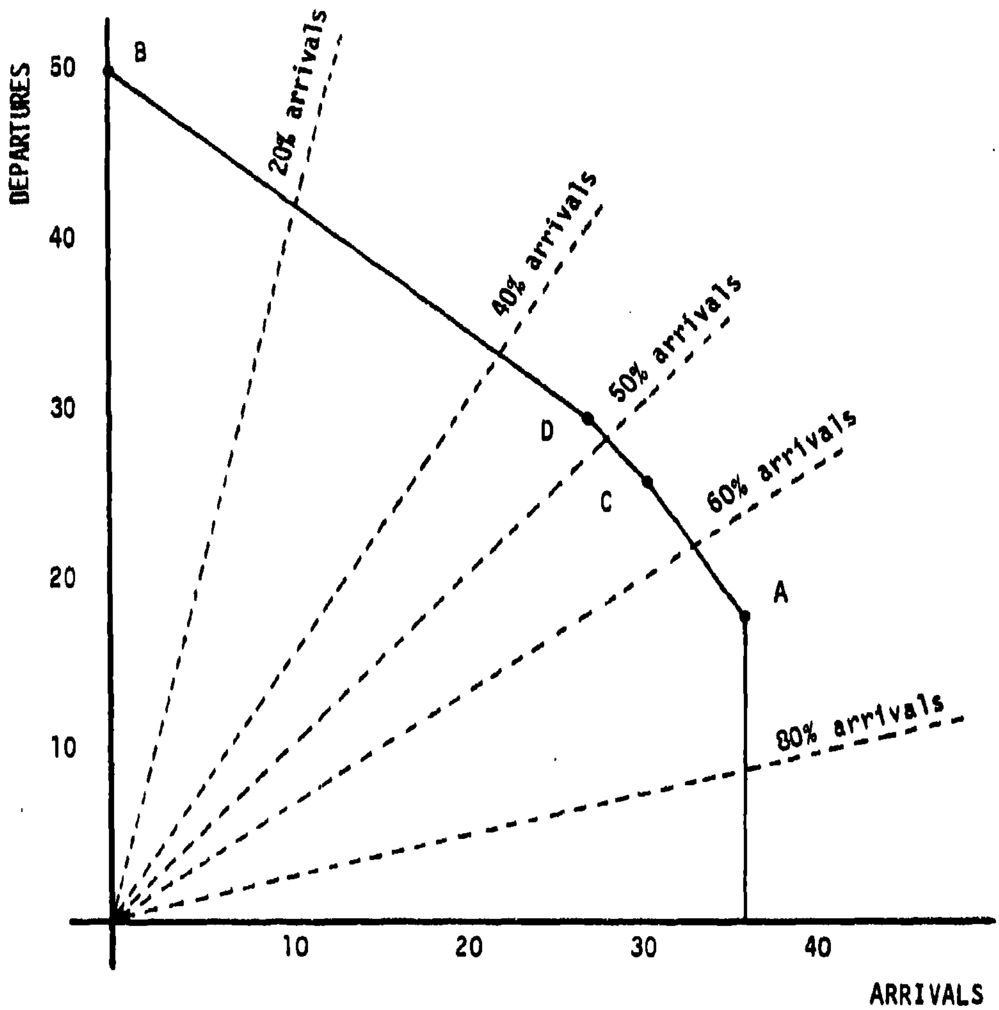


FIGURE 4-2
CAPACITY CURVE ILLUSTRATING MULTIPLE ARRIVAL PERCENTAGES

- o 9999 -- only the arrival-priority capacity is calculated.

- o 8888 -- only arrival-priority and departure-priority capacities are calculated and printed out.

- o 7777 -- all capacity points (arrival-priority, departure-priority, and all intermediate points) are calculated and printed out, with no attempts made to derive a specific proportion of arrivals.

Any one of these can be specified as the first value, without affecting subsequent calculations. However, the program will ignore any special value which is input after the first position. This is done in part to avoid some programming complexity. Also, there is little need for the special codes other than as the first value because the same information (arrival-priority capacity, departure-priority capacity, and relevant intermediate points) is printed out in addition to capacity at the specified arrival percentage.

The capability of storing capacity values and using them to derive the capacity at several arrival percentages has a significant effect on the cost of running the capacity program. Figure 4-3 shows a cost comparison between the original program version and the revised version for a particular example. For this example, the cost of four additional arrival percentages was about 10.5 CPU seconds for the original version, as opposed to about 0.5 CPU seconds for the revised version.

It is tempting to draw other conclusions from Figure 4-3, such as "the original version is cheaper to run with a small number of arrival percentages and a large number of intermediate points." However, this example is illustrative only. The following points must be kept in mind:

- o The original version cannot produce intermediate points and so it is potentially less accurate (see Table 4-1 for a comparison of the results of the two programs).

- o The revised version was also run with the f.e.d. logic activated for a slight increase in cost but greater accuracy.

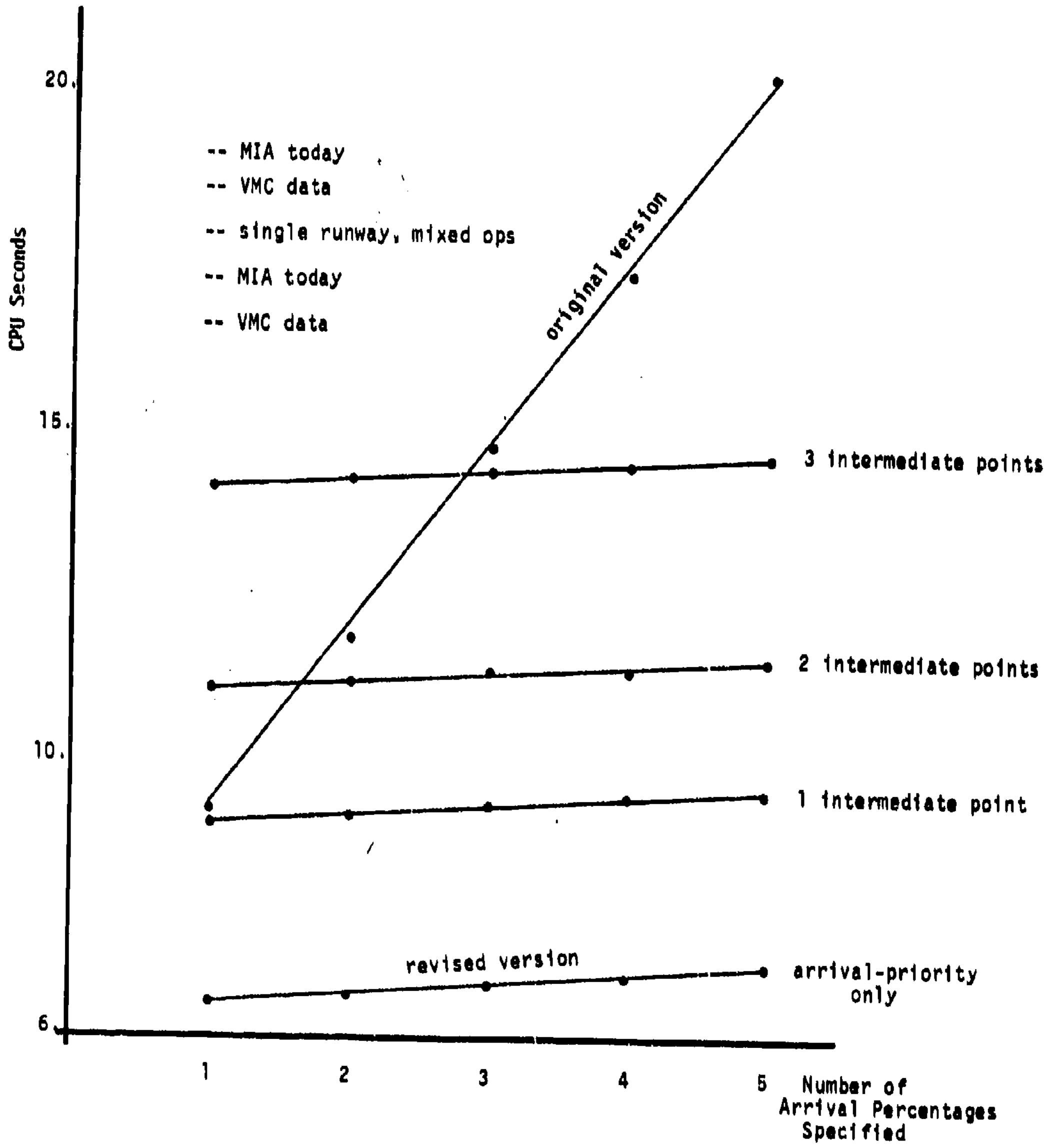


FIGURE 4-3
COST vs. NUMBER OF ARRIVAL PERCENTAGES

TABLE 4-1

COMPARISON OF CAPACITIES AT MULTIPLE ARRIVAL PERCENTAGES

	ARRIVAL PERCENTAGES				
	30	40	50	60	70
ORIGINAL VERSION	49.9	49.9	49.9	49.9	49.9
REVISED VERSION					
ARRIVAL PRIORITY ONLY	51.9 (+4%)	52.6 (+5%)	53.3 (+7%)	54.0 (+8%)	52.3 (+5%)
1 INTERMEDIATE POINT	53.5 (+7%)	54.8 (+10%)	56.1 (+12%)	55.7 (+12%)	53.3 (+5%)
2 INTERMEDIATE POINTS	54.1 (+8%;	55.7 (+12%)	57.0 (+14%)	55.7 (+12%)	52.3 (+5%)
3 INTERMEDIATE POINTS*	54.1 (+8%)	55.7 (+12%)	57.0 (+14%)	55.7 (+12%)	52.3 (+5%)

*3RD POINT WAS IDENTICAL TO THE 2ND AND THEREFORE CONTRIBUTED NOTHING TO RESULTS

- MIA DATA, VMC

-- SINGLE RUNWAY, MIXED OPERATIONS

-- IAT INCREMENT OF 20s

- The cost of obtaining the first arrival percentage includes the costs of loading the program, reading the input file, and printing the results. This cost can be reduced further for the revised version.*

- The multiple percentages were input in ascending order, so that all intermediate points were calculated (in the revised version) before the first capacity value was calculated. Thus a change in the sequence of arrival percentages could decrease the cost of the first result since the intermediate points would not be calculated if they were not needed. The additional cost of the extra arrival percentages would thus be greater, as it would include the cost of calculating the intermediate points.

- The test case was a single runway, mixed operations, VMC. It was run on the MITRE IBM 370/148. Other configurations or other computer systems could change the comparative costs.

4.2 Alternating Arrivals

Several ATC procedures have been changed since the original version of the capacity program was prepared. As the result of one of these changes, it is now a recognized procedure to run dependent alternating arrivals in IMC to parallel runways as close as 3000 feet apart, with as little as 2.0 nmi diagonal separation between arrivals (Reference 1, Para. 797.c). The logic necessary for calculating the capacity of such operations has been added to the program, as subroutine STAGGR.

4.2.1 Description

When alternating arrivals are being conducted, two types of arrival-arrival separations apply -- the diagonal separation between aircraft on different runways, as well as the usual longitudinal separation between arrivals to the same runway. Thus the two previous aircraft can affect the time of arrival at the threshold (not just the one previous aircraft, as for a single runway). In addition, because of a phenomenon we call "shadow spacing," the third aircraft ahead of the current one can affect

* Namely, by creating a MODULE which is easier to load. This has been done since this report was written, and typical running costs have fallen by 2 CPU seconds as a result. The original version of the program was in MODULE form already.

the interarrival times. The many constraints affecting this quartet of arrivals made a single equation expressing interarrival time less attractive than a stepwise calculation of arrival times at the gate and the threshold.

"Shadow spacing" occurs when the spacing between the second and fourth arrivals is determined by the spacing between the first and third in the quartet. (See Figure 4-4.) In this example, aircraft A is a heavy landing on runway 1. Aircraft B, a small bound for runway 2, is 2.0 nmi diagonally behind A. C, a large, is 5.0 nmi behind A on runway 1 because of intrail vortex requirements. The next arrival to runway 2 would be 2.0 nmi diagonally behind C, and therefore 5.0 nmi behind B, a small aircraft which would normally require only a 3.0 nmi separation or less.

In cases of extremely close runway centerline spacing and very small diagonal separations (e.g., 700 feet and 1.0 nmi), it is possible for the fourth preceding aircraft to affect the current arrival. However, these cases are not encountered frequently, and the effect is slight, so the new logic is limited to look only at sets of four, not five, arrivals.

The logic for alternating arrivals is presented in flow chart form in Figure 4-5. The basic technique is to calculate the earliest time for each aircraft to cross the threshold, subject to the constraints of:

- Separation from previous arrival, same runway
- Separation from previous arrival, other runway
- Time to fly from gate to threshold
- Runway occupancy time, previous arrival.

The subroutine also accounts for runway thresholds or approach gates which are displaced relative to each other.

Aircraft #1 is assumed to reach the gate at time zero and proceed unconstrained to the runway. The second aircraft cannot be at the gate to the other runway any earlier than dictated by the diagonal separation requirement (see Figure 4-6):

$$TGT2 = TGT1 + (XSEP + GTDISP) / V(j) \qquad (15)$$

where TGTn = time at gate of the nth arrival
XSEP = projection of the diagonal separation

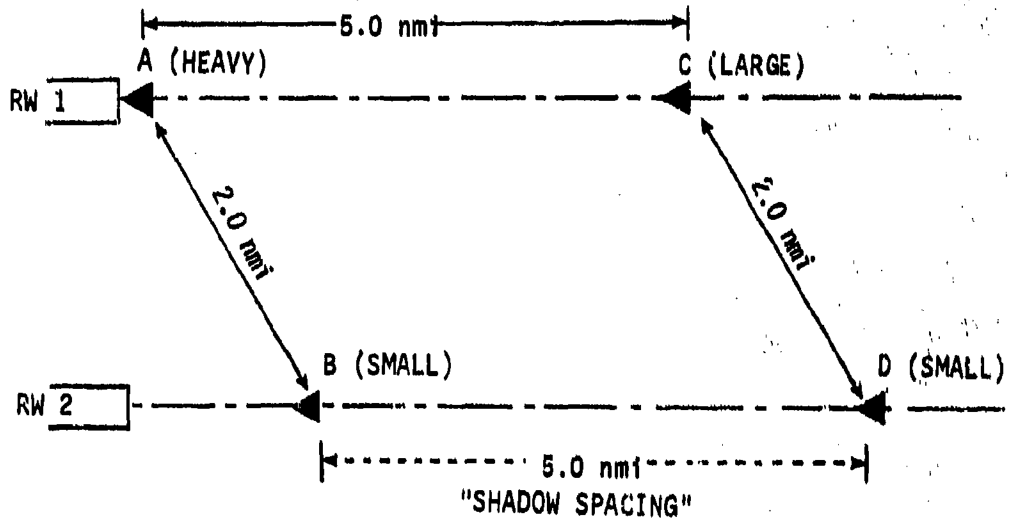


FIGURE 4-4
ILLUSTRATION OF SHADOW SPACING

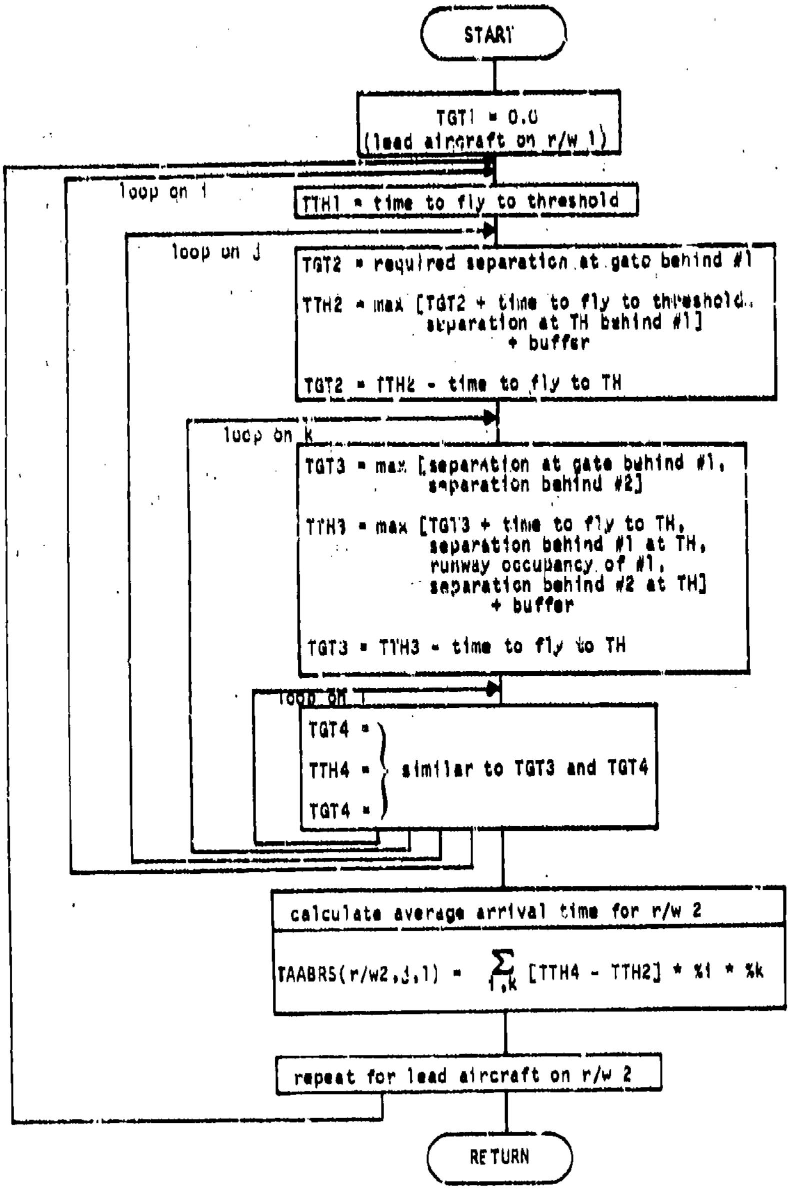
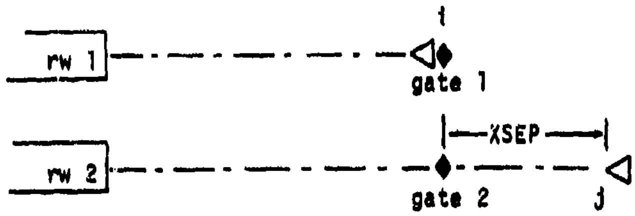
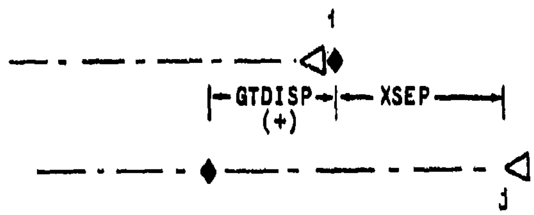


FIGURE 4-5
FLOW CHART FOR ALTERNATING ARRIVALS LOGIC



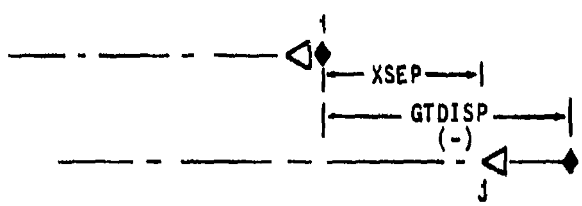
TGT2 = TGT1 + XSEP / V(j)

a) GTDISP = 0



TGT2 = TGT1 + (XSEP + GTDISP)/V(j)

b) GTDISP > 0



TGT2 = TGT1 - (|GTDISP|-XSEP)/V(j)
= TGT1 + (XSEP + GTDISP)/V(j)

c) GTDISP < 0

FIGURE 4-6
TIME AT GATE FOR AIRCRAFT #2

GTDISP = relative displacement of gate #2
V(j) = velocity of j, aircraft #2.

Time at the threshold for j (TTH2) is either the gate time (TGT2) plus time to fly, or the time required by applying the diagonal separation at the threshold, whichever is greater. This latter is calculated as (see Figure 4-7):

$$TTH2 = TTH1 + (XSEP + THDISP) / V(m2) \qquad (16)$$

where TTHn = time at threshold of nth arrival
THDISP = relative displacement of the threshold
V(m2) = a dummy variable, equal to either V(i), the velocity of the lead aircraft, or more often V(j).

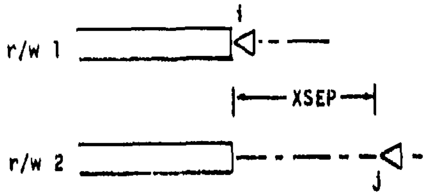
The threshold displacement THDISP, like GTDISP, is a directional quantity: it is positive if the runway 2 threshold (or gate) is displaced ahead of that for runway 1, or negative if it is behind. The only case in which V(i) determines TTH2 is if THDISP is negative and the magnitude of THDISP is greater than XSEP (Figure 4-7(d)). Here the diagonal separation applies when j is at the threshold, since i is still airborne, rather than vice versa.

Once TTH2 is determined, the usual interarrival time buffer is added. The time to fly from the gate to the threshold is then subtracted to obtain the correct value of TGT2.

The threshold-crossing time of the third aircraft (TTH3) may be constrained by either the first aircraft (longitudinal separation or runway occupancy time) or the second aircraft (the diagonal separation). For reasons which will be explained, we have considered these possible constraints separately. TTH31, the time of the third aircraft over the threshold as determined by the first aircraft, is expressed as:

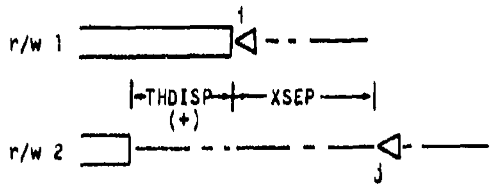
$$\begin{aligned} TTH31 = \max [&TTH1 + DLTAIJ(i,k)/V(k), \\ &TTH1 + AROR1(i), \\ &(DLTAIJ(i,k) + GGAMA)/V(k)] \end{aligned} \qquad (17)$$

where DLTAIJ(i,k) = the minimum longitudinal separation between i and k



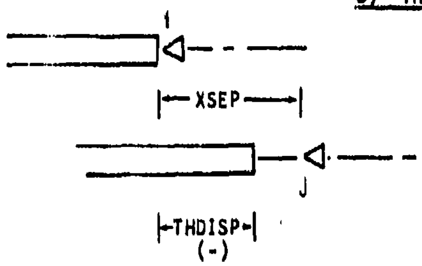
TTH2 = TTH1 + XSEP/V(j)

a) THDISP = 0



TTH2 = TTH1 + (XSEP + THDISP)/V(j)

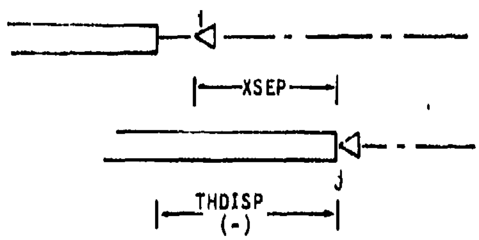
b) THDISP > 0



TTH2 = TTH1 + (XSEP - |THDISP|)/V(j)

= TTH1 + (XSEP + THDISP)/V(j)

c) THDISP < 0, |THDISP| < XSEP



TTH1 = TTH2 + (|THDISP| - XSEP)/V(i)

TTH2 = TTH1 + (XSEP - |THDISP|)/V(i)

= TTH1 + (XSEP + THDISP)/V(i)

d) THDISP < 0, |THDISP| > XSEP

FIGURE 4-7
TIME AT THRESHOLD FOR AIRCRAFT #2

ARORl(i) = the protected runway occupancy time behind arrival i

GGAMA = the gate-threshold distance.

Similarly, TTH32 is the threshold time of 3 as determined by 2, or

$$TTH32 = \max [TGT2 + (XSEP - GTDISP + GGAMA)/V(k), \quad TTH2 + (XSEP - THDISP)/V(m4)] \tag{18}$$

where V(m4) = either V(j) or V(k).

If these equations are compared with those for the second aircraft, it will be noted that the signs of GTDISP and THDISP have been changed. This is because the lead aircraft is now on the other runway.

The actual threshold-crossing time (TTH3) is the maximum of TTH31 and TTH32, plus the interarrival buffer. The time at the gate (TGT3) is determined, as before, by then subtracting the time-to-fly from TTH3.

Similar equations are used to calculate TTH42 and TTH43, and TTH4. The difference between TTH4 and TTH2, averaged over all values of i and k, provides an average time between arrivals j and 1 to runway 2 (termed TAABRS(r/w2,j,1)). The difference between TTH4 and TTH3, averaged over all values of i and j, is termed ALTTAA(r/w1,k,1) -- the average time between consecutive arrivals to the airport, not to the same runway. These calculations are then repeated for the lead aircraft on the other runway.

The standard deviations of TAABRS and ALTTAA are determined as follows. For a single runway, the standard deviation of the interarrival time is an input value, SIGMAA or σ. For alternating arrivals to parallel runways, the standard deviation of TAABRS (the interarrival time between aircraft 4 and 2), termed SIGM42, is equal to σ only if the interarrival time is determined solely by the longitudinal separation between 4 and 2, and the aircraft on the other runway have no effect. (See Figure 4-8(a).)

If the spacing between 4 and 2 is determined by the sum of the average spacings between 4 and 3 and between 3 and 2 (Figure 4-8(b)), the standard deviation equals $\sigma\sqrt{2}$. This is because the variances (σ^2) are additive: the new variance is $\sigma^2 + \sigma^2$ or $2\sigma^2$, and the standard deviation is the square root of this.

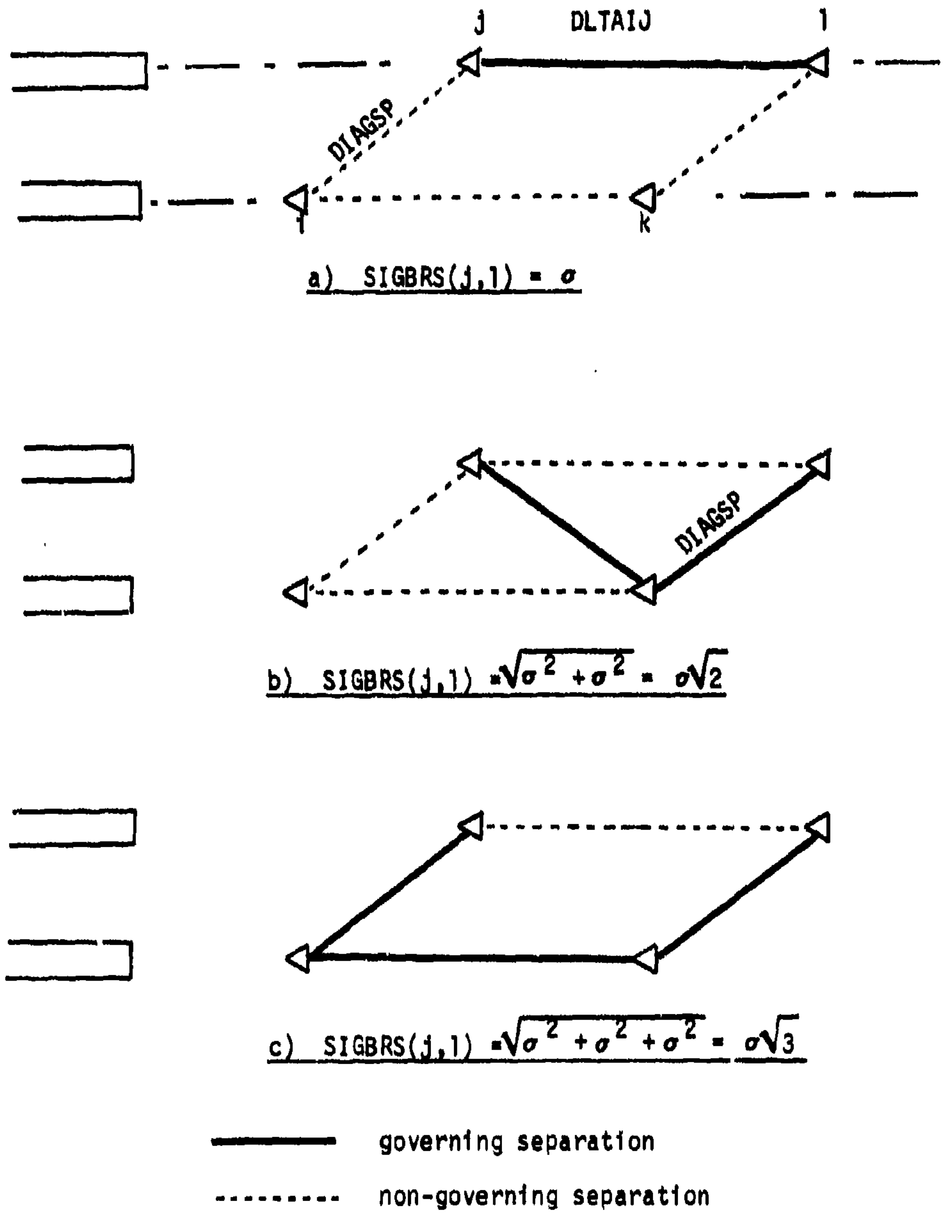


FIGURE 4-8
DETERMINATION OF SIGBRS (j,l)

In the final case, aircraft 3 is constrained not by aircraft 2 but by aircraft 1 (Figure 4-8(c)), and

TTH42 = TTH43 + TTH31 - TTH21.

The standard deviation in this case is $\sigma\sqrt{3}$, by a similar process.

Depending upon the values of i and k, therefore, the standard deviation of TTH4 - TTH2 may be σ, $\sigma\sqrt{2}$ or $\sigma\sqrt{3}$. Which case applies is determined by comparing TTH43 and TTH42, and TTH32 and TTH31; an average standard deviation for all values of i and k is then calculated. This value is termed SIGBRS(r/w2,j,l).

Similarly, the standard deviation of TTH4 - TTH3, termed SIGALT (r/w1,k,l), is determined. For particular values of i, j, k and l, SIGM43 will equal

-- σ if TTH43 > TTH42 (diagonal separation is governing)

-- $\sigma\sqrt{2}$ if TTH43 ≤ TTH42, and TTH32 > TTH31

-- $\sigma\sqrt{3}$ if TTH43 ≤ TTH42, and TTH32 ≤ TTH31.

Once these values for TAABRS, ALTTAA, SIGBRS and SIGALT have been computed, they are returned to TWOPA, the parallel runway subroutine.

To run alternating arrivals, IALT on the NEWRUN line must be set, and data must be input for the diagonal separation (DIAGSP), the distance between centerlines in feet (CLDIST), THDISP, and GTDISP, on line 26 of the input file. The first two are used to compute XSEP as follows:

$$XSEP = \sqrt{(DIAGSP)^2 - (\frac{CLDIST}{6076})^2} .$$

CLDIST is also used to check the appropriateness of operating with alternating arrivals. If CLDIST is less than 3000 feet, the current requirement, alternating arrivals are not run. This can be circumvented, however, by inputting a value of 2 or more for IALT on line 0 (NEWRUN). For subsequent cases in the same run, setting IALT to zero cancels the alternating arrival option. Simply setting IALT to 1 in the first case, without inputting the ALTARR data, will result in an alternating arrival operation with the default values of 2.0 nmi diagonal separation and 3000 feet centerline separations. Figure 4-9 shows the logic involved in these steps.

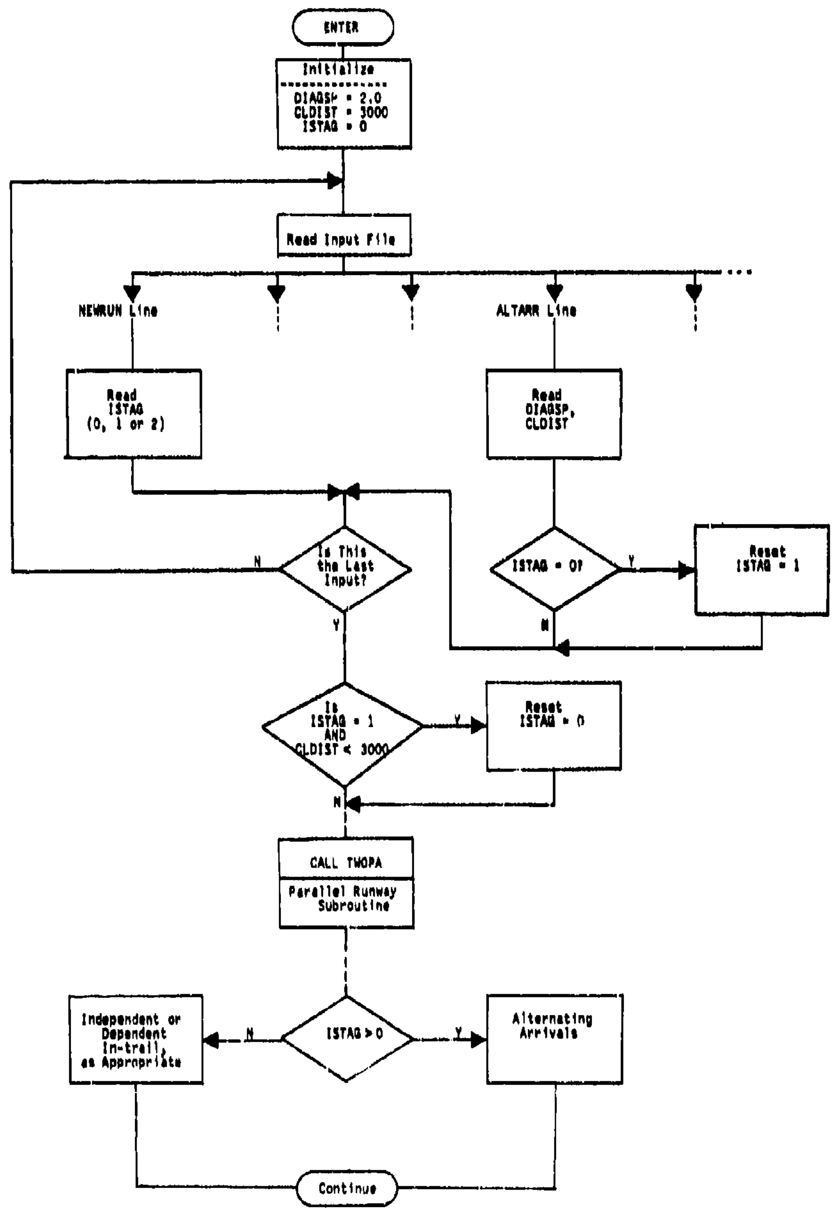


FIGURE 4-9
FLOW CHART FOR DECISION TO RUN ALTERNATING ARRIVALS

Note that alternating arrivals are only an option for medium- and far-spaced parallels in accordance with current procedures. To investigate the capacity of alternating arrivals to close-spaced parallels,

- o input correct value for CLDIST
- o set IALT > 1
- o specify a medium- or far-spaced parallel configuration.

This procedure will not give accurate departure capacities in IMC or MMC if there are departures on both runways (C:B,B). The 2.0 nmi departure/arrival separation between arrivals and departures on different runways, and the departure-departure separation between departures on different runways, are not accounted for. This configuration should be run as (C:B,A) instead. The option to force alternating arrivals is intended only for the experienced user.

Another available option concerns the value used for XSEP. If a negative value of DIAGSP is input, this is a signal to the program to calculate

$$XSEP(i,j) = \sqrt{DLTAIJ(i,j)^2 - (CLDIST/6076)^2}.$$

In other words, DLTAIJ(i,j) is used for both longitudinal and diagonal separation. This can be used (again by the experienced user) to study alternating arrivals to close-spaced parallels under present-day (i.e., no diagonal separations) rules.

4.2.2 Implementation

The original FAA version of the capacity program did not contain the logic for alternating arrival operations. The PMM&Co. version did, however. The logic used was different from that described above in three areas:

- o The logic is more limited (only three different diagonal separations can be specified, present-day longitudinal separations only, etc.).
- o The "diagonal separation" is applied in-trail, parallel to the flight path.
- o Only three aircraft are considered -- "shadow spacing" is not recognized.

The first two differences would be fairly easy to compensate for, but the last would not be so simple. The effect of this difference may be seen in Table 4-2, which compares the costs and results of the two versions.

Both programs agree on the capacity of the single runway, 31.0 arrivals per hour. Running the revised version with zero separation between centerlines most closely approximates the use of in-trail "diagonal" separations in the PMM&Co. version. The differences in the results are due to the use of shadow spacing in the revised version; this has a negligible effect with a 2.5 nmi diagonal but causes the loss of one arrival at 1.5 nmi.

With a 3000 foot runway separation, a 2.0 nmi diagonal is the equivalent of 1.94 nmi intrail. The revised version therefore regains some capacity lost to shadow spacing by applying separations diagonally. The results shown for CLDIST = 3000 feet are approximately the same as for the PMM&Co. version -- sometimes slightly better, sometimes slightly worse.

The costs of running the revised version are comparable as well, although slightly higher: about 4 CPU seconds per run as opposed to 3 CPU for the PMM version. Much of this difference is due to the extra cost of loading the larger revised version. The additional time required by the specialized calculations in the revised version is only about 0.3 CPU seconds per run.

The revised logic does not offer greatly increased capacity or reduced running costs. It is preferable to the logic used previously, however, because it is more flexible, accounts for the effect of different runway centerline spacings, and does not overestimate capacity by neglect of shadow spacing effects.

4.2.3 Application to Special Problems

The alternating arrival logic has been applied to several problem areas in the original model.

In the first, dependent IFR arrivals to medium-spaced (2500 to 4299 feet) parallel runways were originally being modeled as consecutive arrivals to a single runway, with modified arrival separations. However, the separations were not always correctly modified. The revised program uses a variation of the alternating arrival logic for this case: if alternating arrivals (i.e., 2.0 nmi diagonal separation) were not specified by the user, the program runs alternating arrivals with zero runway spacing and a "diagonal"

TABLE 4-2

COST AND CAPACITY COMPARISON OF ALTERNATING ARRIVAL TECHNIQUES

	SEPARATION	PMM&CO. VERSION	REVISED VERSION	
INTRAIL				
	3.0 nmi	31.03 (3.03 CPUs)	31.03 (3.84 CPUs)	
ALTERNATING ARRIVALS			0' CLDIST	3000' CLDIST
	2.5 nmi	35.23 (3.05 CPUs)	35.21	35.67 (4.4 CPUs)
	2.0 nmi	40.66 (2.91 CPUs)	40.51	42.21 (3.90 CPUs)
	1.5 nmi	47.23 (2.93 CPUs)	46.17	47.03 (4.07 CPUs)

separation equal to DLTAIJ(1,1), normally the smallest arrival spacing. The zero runway spacing implies that this spacing (3.0 nmi presently) is applied intrail.

If the runways are spaced far enough apart, some capacity can be gained by applying the separations diagonally rather than intrail. This can be investigated by specifying alternating arrivals and inputting the actual diagonal separation (e.g., 3 nmi) and the actual runway separation. For runway separations below 2500 feet, where vortex buffers must be added to the arrival spacings, the same separation matrix will be used for longitudinal and diagonal spacing if the user inputs a negative value for the DIAGSP (see 4.2.1 above).

In VMC such close-spaced parallel runways are independent except for such vortex effects. The original program calculated the vortex-free arrival capacity and the vortex-constrained capacity, and took a weighted average based on the proportion of heavy aircraft in the mix. This could only be an approximation, however, because the same mix of vortex-producing and non-vortex aircraft was used in each case.

An attempt was made to use the alternating arrival logic in this situation. The program was modified so that the diagonal separation applied only when the aircraft on the other runway was a vortex-producer; otherwise, the trail aircraft was constrained only by the longitudinal separation required behind the aircraft ahead of it. A comparison case was run using ATL data, and identical arrival capacities were computed by the two different methods. It was decided then that adapting the alternating arrival logic to this case was not necessary and no changes were made. Further investigation may show benefits in other cases, or significantly lower costs, which would make the alternating arrival logic preferred.

The alternating arrival logic has also been applied to certain triple runway configurations in IMC. For example, model 3-7 (N: B,B,A) has mixed operations on runways 1 and 2, which are close-spaced, and arrivals-only on runway 3 (spaced 2500-3500 feet from 2). In IMC, it is assumed that alternating arrivals are operated to the outer runways, and departures only on the center runway. Since runways 2 and 3 are more than 2500 feet apart, the departures on 2 are only dependent upon the arrivals to runway 1. Runways 1 and 2 are therefore treated as a standard dual-lane pair, with the exception that TAABRS(r/w1,i,j) is used for the average interarrival time, and SIGBRS(r/w1,i,j) is used for the standard deviation of the time between arrivals i and j.

Model 3-17 (C:A,D,A) is slightly different. Since here runways 2 and 3 are close-spaced (700-2500 feet), departures on 2 must be inserted between an arrival on runway 1 and an arrival on runway 3. The average time ALTTAA and standard deviation SIGALT are therefore used to compute departure capacity. The number of departures is computed first for the lead arrival to runway 1 and then for the lead aircraft on runway 3; the results are then combined appropriately to give departure capacity. To accommodate these features, the appropriate changes were made to subroutine DUAL.

4.3 New Intersecting Runway Models

4.3.1 Need for Departure-Departure Model

The original capacity program considered two operating modes on intersecting runways: arrivals on one, departures on the other (A,D), and arrivals on one, departures on both (B,D). Arrivals to both runways would never occur in the real world, so there is no reason to model it, but departures on both are feasible. New logic to compute departure-departure capacity was added to the revised program as Model 6-1 (D,D), subroutine TWOIN.

Departures on both intersecting runways would not be the primary configuration for an airport since no arrivals occur, but it is necessary at times for the calculation of departure-priority capacity. Model 6-3, for example, with arrivals on one and departures on both, would revert to intersecting departures to obtain the departure-priority capacity, as would some of the complex configurations.

The advantages of intersecting departures over a single departures-only runway are twofold:

- o departure runway occupancy is less critical
- o wake vortex separations are less restrictive.

The first is obvious: the departure on the second runway does not have to be held until the first departure is clear of the runway, but just until it clears the intersection.

The second advantage stems from a difference in where the vortex separations are applied. A non-heavy cannot be released sooner than the vortex separation (120s today) after a heavy departure starts to roll. With intersecting departures, if the heavy departure is not airborne at the intersection, vortex separation (120s) does not apply to the intersecting departure; if it is

airborne, then the intersecting departure cannot pass through the intersection less than 120s after the heavy. However, if the intersection is not equidistant from the two thresholds, this could mean less than 120s between departure release times, and so capacity would benefit.

4.3.2 Description of Logic

On a very general level, operations on the departure-departure runway and alternating arrivals to parallel runways are similar, and so is the model logic for each. The interval between departures depends not only on the required separation behind the previous departure on the same runway, but also on the separation needed at the intersection due to wake vortex behind the last departure on the other runway. A set of five departures must be considered, in part because the smallest inter-departure time is less than reasonable inter-arrival times with alternating arrivals.

Model 6-1 requires that line 11, TWOIN, must be in the input file. This line contains the variables:

- o IAX, the airborne intersection indicator (0 if not airborne, 1 if airborne)
- o TXI(k,m), the average time for departure k on runway m to clear the intersection.

Given these, we can compute the times for each departure to be released as constrained by:

- o TXI of the previous departure
- o DDSR, the departure-departure separation, behind the previous departure on the same runway
- o DDSR at the intersection, if the previous departure (other runway) is heavy and airborne at this intersection.

A flow chart of this logic is shown in Figure 4-10.

This logic assumes strict alternation of departures on the two runways. In certain cases, it might be possible to insert an additional departure into the stream without affecting the other departures. The program calculates the number of additional

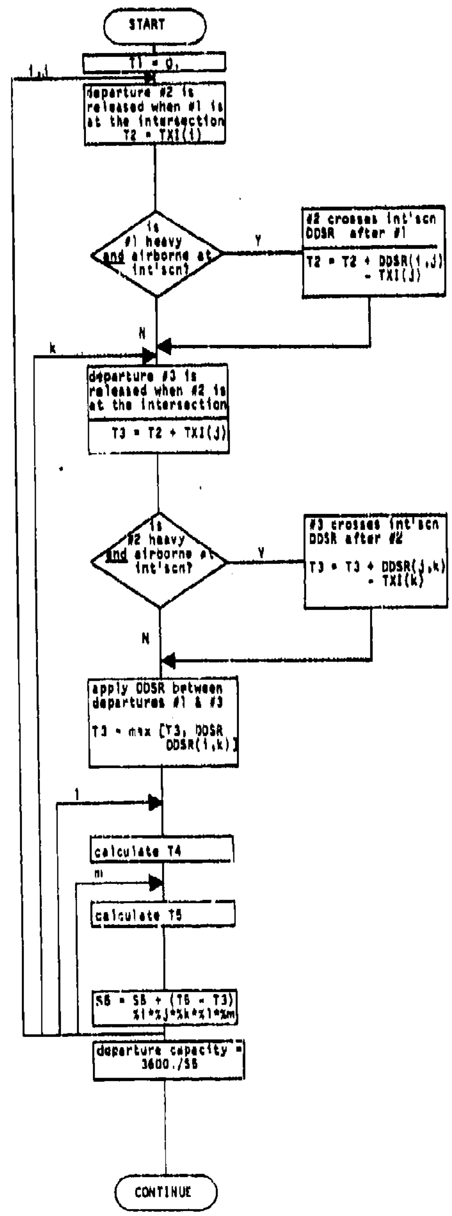


FIGURE 4-10
FLOW CHART FOR INTERSECTING DEPARTURES LOGIC

departures which might occur in this way and prints it out as "ADDITIONAL CAPACITY IF DEPTS. NOT STRICTLY ALTERNATING." This number is not included in the departure capacity; it is usually small.

After this intersecting departure capacity is calculated, the single-runway departure capacity is computed. The larger capacity is returned to the main program as the capacity of the configuration. (See Table 4-3.)

If the weather is IMC, we assume that intersecting departures would not be run, and only the single-runway capacity is calculated.

4.3.3 Model 6-3 -- Arrivals on One, Departures on Both

The capacity program includes one intersecting runway configuration with arrivals to one runway and departures on both (Model 6-3). The capacity of this configuration is taken to be the greater of either:

- arrivals on one, departures on the other (Model 6-2) or
- mixed operations on a single runway (Model 1-3).

In other words, one or the other departure stream is always dropped.

It is possible to conceive of a configuration with a short intersection in which a greater capacity could be attained by running departures on both runways. As long as the intersecting departure cleared the intersection before the arrival exited the runway, it would not affect the departure on that runway.

Although the need for such logic is recognized, such logic has not yet been developed. It would be difficult, if not impossible, to combine the probabilistic departure logic of Model 6-2 with the more deterministic logic of 6-1. A partial combination might be feasible, although it would not necessarily give the greatest capacity. This alternative might assume that the first departure was on the intersecting runway, and all others were on the main runway. The time at which the second departure is released would no longer be DDSR after the first departure, but would be either when the first departure cleared the intersection (with vortex buffer, if needed) or when the arrival exited the runway, whichever came first.

TABLE 4-3

COMPARISON OF INTERSECTING AND SINGLE-RUNWAY DEPARTURES

INTERSECTION DISTANCE		AIRBORNE?	INTERSECTING DEPARTURES	SINGLE-RUNWAY DEPARTURES
R/W 1	R/W 2			
1000 FT.	3000 FT.	NO	86.2	53.7
5000 FT.	6000 FT.	YES	52.7	53.7
8000 FT.	9000 FT.	YES	44.4	53.7

-- 1% A, 13% B, 73% C, 13% D

-- PRESENT DAY ATC

-- VMC

In view of the anticipated complexity of this logic and the low priority for the change, little work has been done so far in this area. The coding for Model 6-3 has been modified, however, so that it prints out a message telling the user whether the intersecting or single runway provided the greater capacity.

5. OTHER MODEL MODIFICATIONS

In addition to the major logic changes and the new capabilities which were added to the capacity program, many changes were made to correct errors, reduce running time, or improve program input/output. Most of the changes will be discussed in the following pages, grouped by the subroutine in which they occurred:

- SINGLE -- single runway
- TWOPA -- two parallel runways
- TWOIN -- two intersecting runways
- the complex configurations (THREPA, FOUINB, etc.)
- PROB -- normal distribution
- CRAIG -- interactive input.

5.1 Single Runway

A modification to the SINGLE subroutine was needed to correct an observed anomaly: as ceiling and visibility decreased, capacity increased. A large jump occurred as weather transitioned from VMC conditions to IMC conditions (at 1000 ft. ceiling, 3 mi. visibility), as shown in Figure 5-1. It remained at this level and then rapidly declined to a final stable value.

In VMC, a departure can be released if it will takeoff before the next arrival crosses the threshold. In IMC, the simultaneous occupancy rule is effectively superceded by the 2.0 nmi departure/arrival requirement: the departure cannot be released if the arrival is within 2.0 nmi (DLTADA) of the threshold. In marginal conditions, the program assumes that visual separation can be applied between the departure and the arrival after the arrival is within visual range of the runway. This means that the departure stream must stop when the arrival is 2.0 nmi from the threshold, but can resume when it is EPSILN from the threshold, where EPSILN is either the slant range visibility or the distance at which the arrival breaks through the ceiling, whichever is less. A departure is permitted during the interval from EPSILN to the threshold only if it can liftoff in time, and if it does not interfere with other departures.

The jump in capacity was traced to two faults in the logic for calculating these additional departures. In the original version,

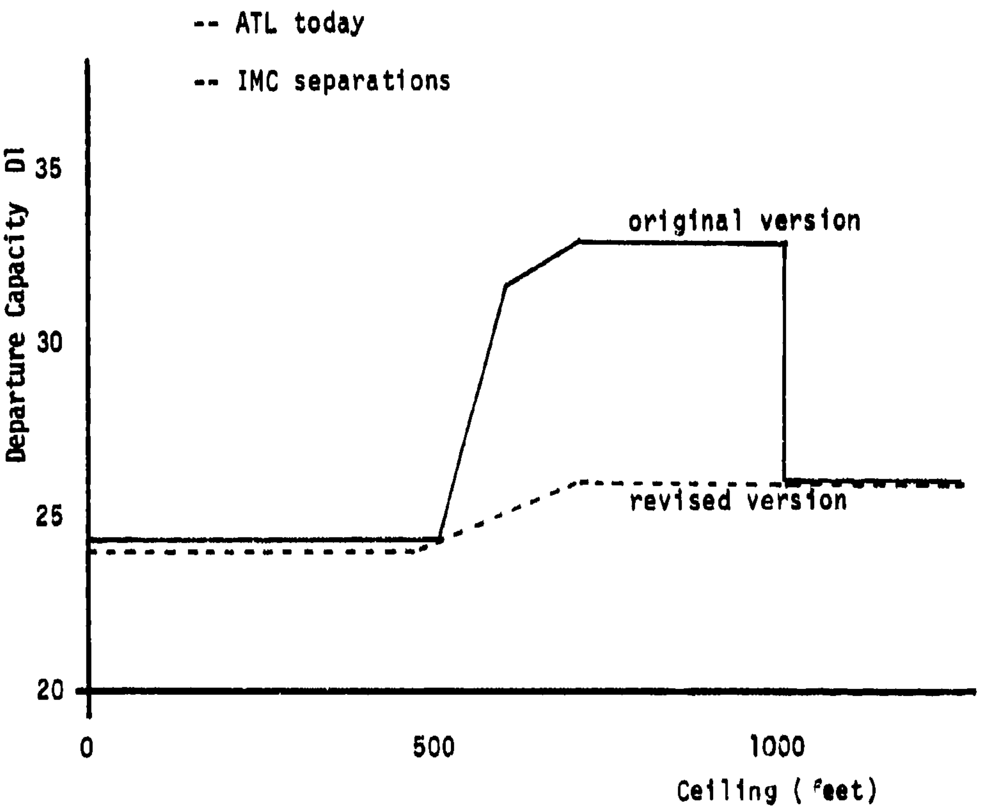


FIGURE 5-1
CEILING/VISIBILITY PROBLEM, SINGLE RUNWAY

an extra departure was allowed if:

$$DROR2 \le \text{time to fly EPSILN} \quad (19)$$

where DROR2 = the departure runway occupancy time.

This has been changed in the revised version. The criterion for allowing extra departures is now (see Figure 5-2):

$$\max \begin{Bmatrix} DROR2(k'), \\ DDSR(k',nm) - ARBAR(j) \end{Bmatrix} \le \min \begin{Bmatrix} \text{time to fly EPSILN}, \\ (\overline{TAA}(i,j) - ARBAR(i) - DDSR(k,k') \end{Bmatrix} \quad (20)$$

where k' = the extra departure

k = the previous departure

nm = the next departure

DDSR(k',nm) = the required time between departures k' and nm

i = the previous arrival

j = the next arrival

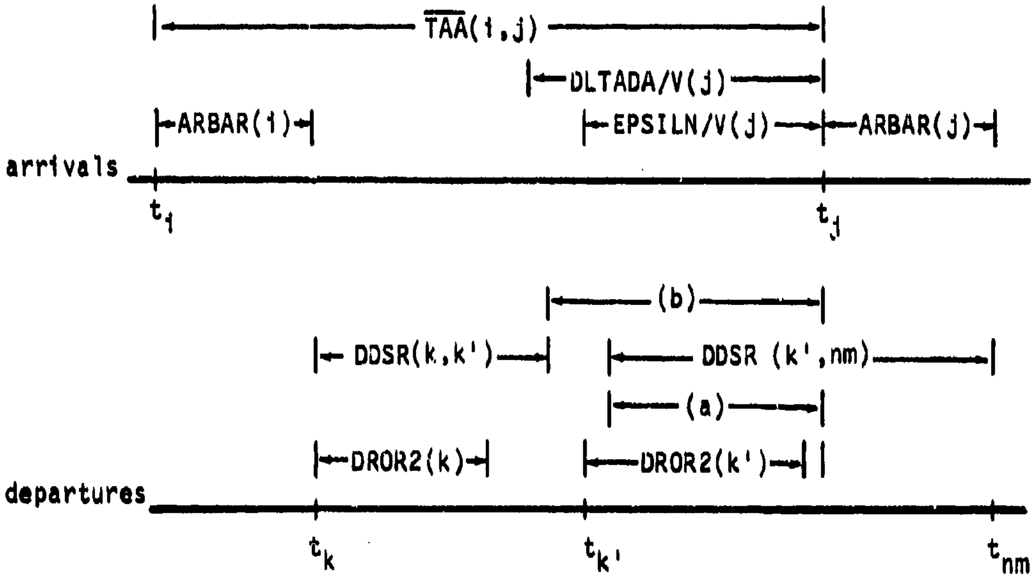
TAA(i,j) = the average time between arrivals i and j

ARBAR(i) = the average runway occupancy time of arrival i.

The time required for the extra departure is on the left side of the above equation. The second term (DDSR(k',nm) - ARBAR(j)) refers to the need to avoid influencing the next departure nm, which is assumed to be released the instant that arrival j clears the runway.

The right side of the equation is the time available for the extra departure. The second term here represents the required separation behind the previous departure k. Only if the time required is less than or equal to the time available can the additional departure be released.

The results of this revised logic are also shown in Figure 5-1. The departure capacity is higher in VMC, remains constant as the ceiling decreases, and then declines smoothly to the IMC level. The new IMC departure capacity differs slightly, probably due to changes which were made to subroutine PROB (see Section 5.5); the



(a) = DDSR(k',nm) - ARBAR(j)

(b) = $\overline{\text{TAA}}$(i,j) - ARBAR(i)-DDSR (k,k')

FIGURE 5-2
TIME AXIS DIAGRAM OF ADDITIONAL DEPARTURE LOGIC

VMC capacity is higher because the revised program sets DLTADA, the departure/arrival separation, equal to zero in VMC regardless of the value which was input. This is necessary because the program uses DLTADA, as well as the ceiling and visibilty, to decide whether conditions are VMC, MMC, or IMC. As the flow chart in Figure 5-3 indicates, if IMC is desired, a non-zero DLTADA must be input and ceiling and visibility must be such that EPSILN is less than DLTADA. Not setting DLTADA to zero was the second fault of the original program: extra departures were being calculated although they conflicted with normal operations.

5.2 Parallel Runways

5.2.1 Near Spaced Runways

In the original program version, parallel runways were classified according to the distance between centerlines, as follows:

- o far (centerline separation 4300 ft or more)
- o medium (3500-4299 ft)
- o near (2500-3499 ft)
- o close (700-2499 ft).

These categories were based upon the rules for independent arrival/arrival and arrival/departure operations in IMC and for wake vortex dependence.

Since the original program was written, some of the rules have changed. Whereas 3500 feet was previously required for simultaneous arrivals and departures on parallel runways, only 2500 feet is required today. This rule was the only reason for separate "near" and "medium" categories. Near-spaced parallels are characterized by dependent arrival/departure operations, but no vortex effects between runways. Medium spaced parallels allow independent arrivals and departures, and no vortex effects. The reduction in the centerline separation required for simultaneous arrivals and departures, to the same value required for independence from vortex effects, eliminates the need for a special "near" category.

If the user of the revised program specifies a near-spaced runway (models 2-13 through 2-18) the program will run the appropriate medium-spaced configuration instead. This is accomplished by simply subtracting 6 from the ISTRGY input. Model 2-14, near-spaced parallels with arrivals on one and departures on the other (N:A,D), therefore becomes 2-8, the medium-spaced configuration

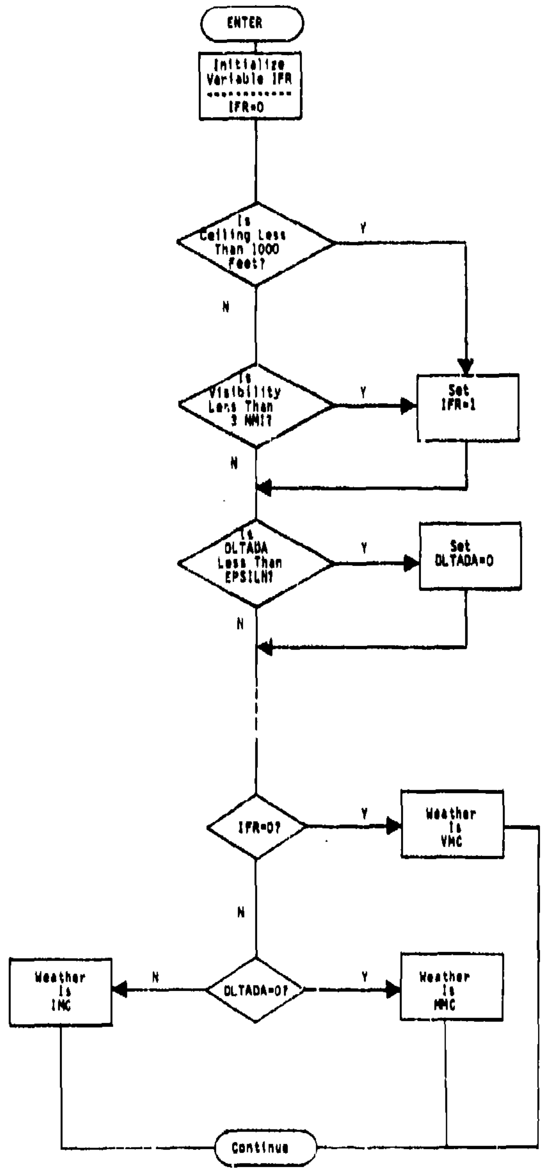


FIGURE 5-3
FLOW CHART OF VMC, MMC, IMC DECISION

(M:A,D). The logic for such near-spaced configurations has not been eliminated from the program, it is just not directly accessible any longer.

The "near" category is still employed for triple runway configurations. The model logic for triples includes the convention that runways 1 and 2 are close-spaced (700-2499 feet). In addition, the spacing between the outer pair (1 and 3) is assumed to depend upon the spacing between 2 and 3, as follows:

- o if 2 and 3 are close, 1 and 3 are medium
- o if 2 and 3 are "near," 1 and 3 are medium
- o if 2 and 3 are medium, 1 and 3 are far
- o if 2 and 3 are far, 1 and 3 are far.

In this case, the "near" spaced runways are operated according to the rules for medium-spaced runways; the only difference between the second and third cases above is whether simultaneous IFR arrivals are allowed (1 and 3 are far-spaced) or not (1 and 3 are medium-spaced).

5.2.2 Runway Occupancy on Parallel Runways

If parallel runways are less than 2500 feet apart, the program will allow simultaneous arrivals in VMC but not in IMC conditions. In MMC or IMC, the runway pair is operated as a single runway.

The original program assumed that, in this case, all arrivals were conducted to runway 1. This may or may not be true, but it would make a difference in capacity only if there were a substantial difference in runway occupancy times between the two runways.

The logic in the revised program has been modified so that the arrival capacity is computed for both runway 1 and runway 2, and the maximum of the two is the value returned. If the user knows that one runway in particular is used in IMC, this configuration should be modeled as a single runway in those conditions and not as a pair.

5.3 Intersecting Runways -- Constant Time to Clear Intersection

The time required for arrivals and departures to clear the runway intersection is an important factor in determining the capacity of crossing runways. There is a question, however, about the value to be input.

The original program version seemed to assume that the time to clear the intersection was constant: the standard deviation of such times was never considered. This would be appropriate if the clearance times to be input were "protected times" or "never-exceed times," which represent worst cases rather than actual performance. If average values are to be used, however, the standard deviation should also be a factor in the calculations. This has been done in the revised program version. SIGMAA, the standard deviation of the interarrival time, has been replaced in the TWOIN logic, where appropriate, by:

$$SIGMAA^2 + SIGAI^2 + SIGDI^2$$

where: SIGAI = standard deviation of arrival time-to-clear intersection

SIGDI = standard deviation of departure time-to-clear intersection.

Values of SIGAI and SIGDI can be input on the TWOIN line, line 11 of the input file. The default values are zero, reflecting the original program usage. However, the user now has the option to use the time-to-clear as either an average or a protected time.

5.4 Complex Configuration

The capacity program can calculate the capacity for configurations of one, two, three or four runways, parallel or non-parallel, intersecting physically or beyond the threshold. Most of these configurations are modeled by the program as combinations of single runways, parallel pairs and intersecting pairs. In this report, these three configurations will be termed "simple configurations," and all others will be referred to as "complex configurations."

5.4.1 Revisions to Default and K-models

For complex configurations, as with the simple configurations, the program computes the arrival priority capacity first. This will be the sum of the arrival priority capacities of one or more simple configurations. For example, three parallel runways with mixed operations on each are assumed to be independent in VMC. The arrival priority capacity of the complete configuration would be the sum of the arrival priority capacities of the three individual runways, or

Cap 3-1(C:B,B,B) = Cap 1-3(B1) + Cap 1-3(B2) + Cap 1-3(B3)

where Cap 1-3(B1) = the capacity of mixed operations on runway 1, calculated using model 1-3.

The arrival priority capacity of a complex configuraton is the sum of the capacities of the "default" models. In the above example, the default models are 1-3 (single, mixed ops) for each of the three runways.

The departure-priority capacity of the configuration is determined by evaluating the capacity of a similar runway configuration, but with conflicting arrival streams removed. For the present example, this would mean

Cap 1-2(D1) + Cap 1-2(D2) + Cap 1-2(D3)

or departures-only on each runway. The structure of the program prevents this combination of simple configurations from being specified directly; instead, each model has an associated K-model, whose default models provide the departure-priority capacity for the first model. The K-model for 3-1(C:B,B,B) in VMC is 3-28 (C:D,D,D).

The relationship between the original model, the default models and the K-model is illustrated, for this example, in Figure 5-4.

The correct choice of default models and K-model for each complex configuration depends primarily upon the weather (VMC, MMC or IMC), as this determines whether VFR or IFR procedures are to be followed. Sometimes the same default model is used regardless of the weather, with the weather-related decision occurring within the default model itself. Sometimes the choice of default models is based upon other factors as well, such as the angle between two open-V runways.

In some cases a subjective decision by the programmer was required before the default model could be specified: for example, in IMC, if only two parallel arrival streams are allowed, which two runways will receive arrivals? Does this choice affect capacity? In each such case, every attempt was made to maximize capacity, subject to the relevant ATC procedures.

The default models and K-models in the original program have been thoroughly reviewed, and occasionally changes were made. In some cases the specified default or K-model did not maximize capacity; in others, proper ATC procedures were contradicted. An example of the first was the K-model for 3-1(C:B,B,B) in VMC -- originally model 2-20(C:A1,D2) was specified rather than 3-28(C:D,D,D). An example of the second fault, incorrect application of ATC procedures, occurred with model 3-6(F:A,D,B) in IMC (see Figure 5-5). The default models originally were:

1-1(A1) + 1-2(D2) + 1-3(B3),

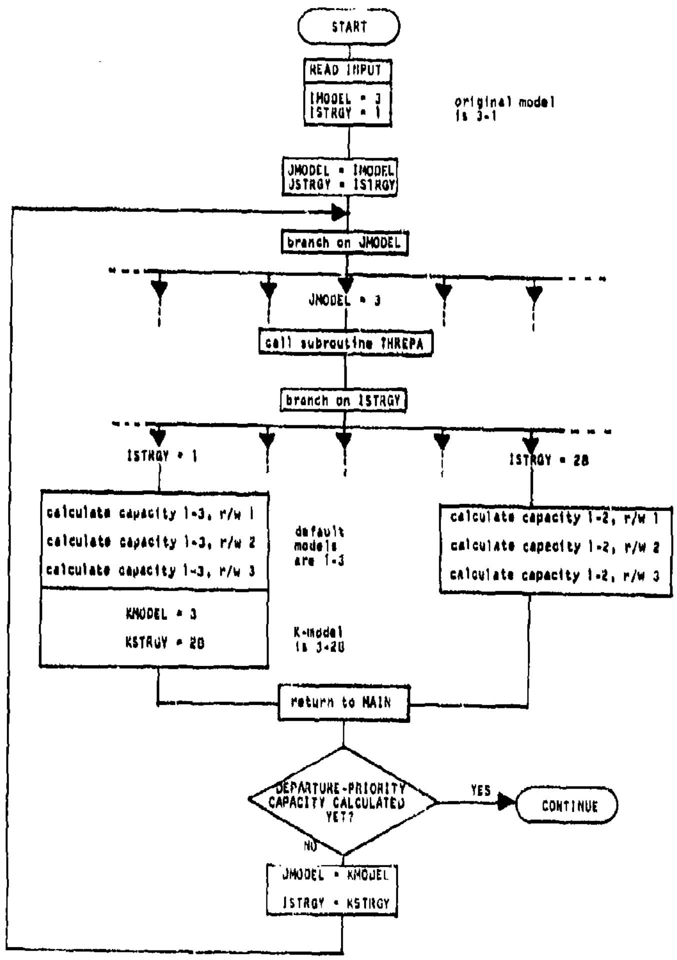


FIGURE 5-4
RELATIONSHIP BETWEEN ORIGINAL, DEFAULT, AND K-MODELS

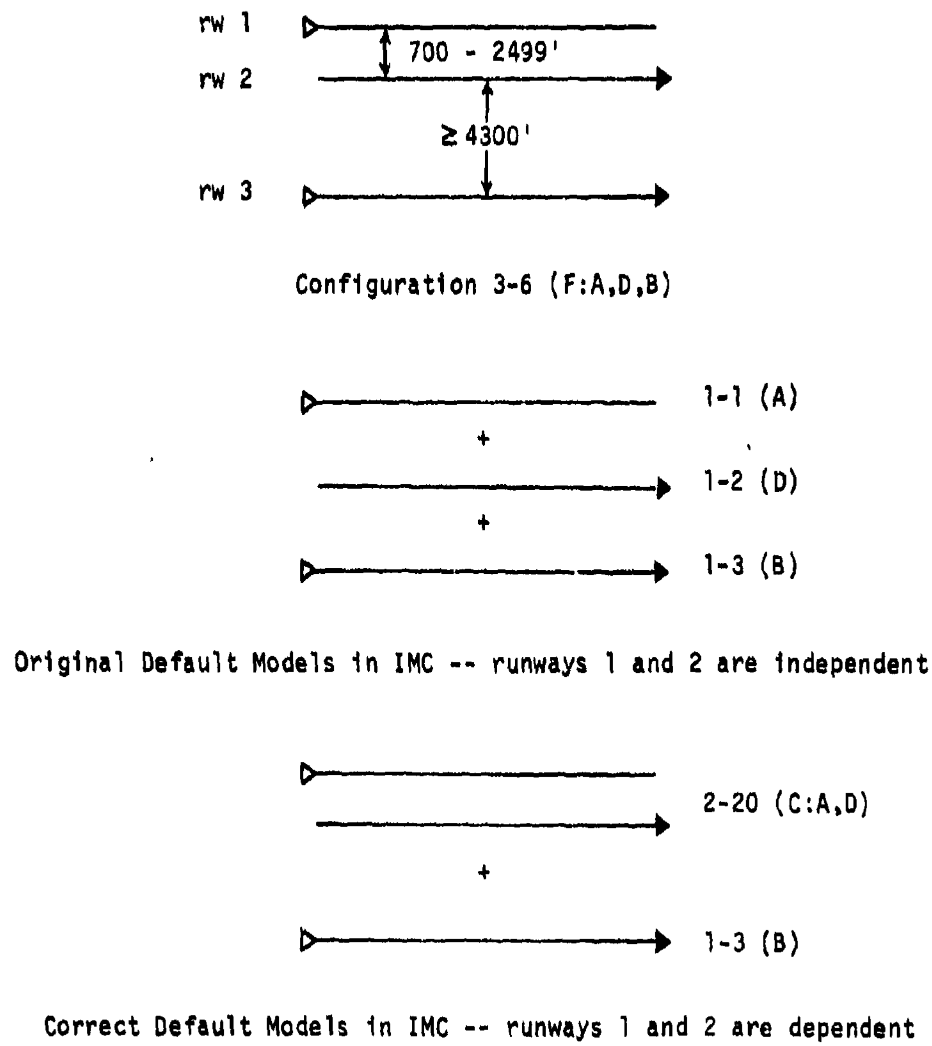


FIGURE 5-5
ILLUSTRATION OF REVISED DEFAULT MODELS, CONFIGURATION 3-6

but in IMC, the departures on runway 2 should be dependent upon the arrivals to runway 1, which is close-spaced and not independent. In the revised program, the default models are:

2-20(C:A1,D2) + 1-3(B3).

A complete listing of the default and K-models for all the configurations, simple and complex, is given in Appendix D.

This review showed the need for some new models, which were then designed and added to the program. These are indicated in Appendix D. One such new model is the default in IMC for 3-25(C:D,A,D). The departures on runways 1 and 3 are both dependent on the arrivals to runway 2, but they are independent of each other. The new model calculates capacity by running 2-20(C:A2,D1) and then 2-20 (C:A2,D3), adding the two departure capacities.

A more complicated new model is the IMC default for 3-5(N:A,D,B). The arrivals on runway 1 and runway 3 are dependent, being medium spaced; departures on runway 2 are dependent on arrivals to runway 1 (close-spaced), but are independent of the departures on runway 3 (near-spaced -- 2500-3499 feet). This capacity is calculated by first evaluating 2-7(M:A1,A3), which returns values for TAABRS (n,i,j) -- the average interarrival time between i and j to runway n, with alternating arrivals. The values of TAABRS (runway 1,i,j) are then used by the dual-lane runway logic (subroutine DUAL) to derive the departure capacity on runway 2, and TAABRS (runway 3,i,j) is used by 1-3(B3) to compute the number of departures on runway 3.

5.4.2 Modifications to Subroutine CONDAB

Several of the complex configuration subroutines refer to subroutine CONDAB before calculating capacity. If the runways are open-V or intersect beyond the threshold (i.e., non-parallel but not physically intersecting), the angle between the runways (θ) and the distance between the thresholds (d) affects the manner in which the runway can operate.

In its original form, CONDAB seemed to use the non-radar IFR procedures in the Controller's Handbook (Reference 1) to determine the dependencies between arrivals and departures. CONDAB has been rewritten to use the IFR radar rules (Paragraphs 742 and 744 of the Controller's Handbook) instead.

These rules indicate that for $\theta < 15^\circ$, the two diverging open V runways are considered as two parallels. For $\theta \geq 15^\circ$ and runway edges not touching, the two runways are independent with respect to AD, DA and DD operations, but AA are dependent. AA operations are

dependent because the arrival streams or the missed approach paths cross in mid-air; AD, DA and DD operations are independent on diverging runways but dependent on converging runways.

CONDAB has been modified to incorporate the IFR radar rules. The value of the indicator variable IR is determined as follows:

IR = 0 for $\theta < 15^\circ$ and $700' \leq d < 2500'$ (close "C" parallels)

IR = 1 for $\theta < 15^\circ$ and $2500' \leq d < 4300'$ (medium "M" parallels)

IR = 2 for $\theta < 15^\circ$ and $d \geq 4300'$ (far "F" parallels)

IR = 3 for $\theta \geq 15^\circ$ and $d > 0$.

IR = 0 therefore implies complete dependence of AD, DA, DD and AA operations. If IR = 1, AA are dependent but the others are not, because of the rules which apply to parallel runways. If IR = 2, even AA operations are independent. If IR = 3, AA operations are dependent because the arrival streams or the missed approach paths cross in mid-air; AD, DA and DD operations are independent on diverging runways, but dependent on converging runways.

Examination of the Controller's Handbook (Reference 1) revealed no mention of wake vortex constraints on open-V operations. It was, therefore, assumed that operations on the different runways were vortex independent if IR = 3. If $\theta < 15^\circ$ (IR < 3), the runways were treated as parallel; operations were vortex dependent if $d < 2500$ feet (IR = 0), the usual criterion for parallel runways.

The value of IR is returned to the subroutine which called CONDAB, where it then controls the branching of program logic.

5.4.3 Default Models -- Missing Input

Occasionally, the default models for certain configurations will require input that the original configuration does not normally need. If this data is not input, one of two possibilities occur:

- either default values or zero values will be used instead of the correct input, or
- if values for these items have been entered for a previous case, the same input values will be used, perhaps inappropriately.

This situation can arise, for example, in the case of an open-V runway. An open-V runway (non-parallel runways, but no physical runway intersection) will be modeled as an intersecting runway if

- o weather is IMC (all IFR radar rules apply)
- o $\theta \geq 15^{o}$ and $d > 0$
- o converging arrivals and departures.

This describes models 5-4 (CV:D,A) and 5-5 (CV:B,A) in IMC, IR = 3 (see Appendix D). The arrivals and departures must be dependent because of the possible conflict between a departure and a missed approach.

However, treating this as an intersecting runway requires specifying appropriate values for the departure/arrival separation (DICBR) and the arrival/departure requirement (ADSR). DICBR is the required distance of an arrival from the threshold when the departure is released, based on the arrival being 2 nmi from the intersection when the departure clears the intersection. The geometry of the runways leads to a proper value of DICBR which is fairly large -- generally more than 2.0 nmi. Similarly, ADSR for an open-V runway could be large. ADSR is the time it takes the arrival to clear the intersection; the departure can be released when the arrival clears the intersection or exits the runway, whichever comes first. A zero value of ADSR would mean that the departure could be released as soon as the arrival crossed the threshold, which is probably too lax a criterion. In reality, the departure would probably not be released until the arrival was committed to a landing (about 10s after crossing the threshold), or even not until it exited.

If ADSR and DICBR have been specified for a previous case in the same run, the same values will be used for the open-V case -- even if they are extremely inappropriate.

There are three possible "solutions" to this problem:

- o Educate the user so ADSR and DICBR are input when needed
- o Add ADSR and DICBR inputs to the open-V input line
- o Add non-zero default values for ADSR and DICBR to the program.

Program modifications have been made to incorporate both the second and the third solution .

Two items, ADSRX and DICBRX, have been added to line 10 (OPENV) of the input file. If non-zero values are input, then

$$\left.\begin{array}{lcl} ADSR(i,j,n2) & = & ADSRX \\ DICBR(i,j) & = & DICBRX \end{array}\right\} \quad \text{for all values of } i,j$$

where n2 is the departure runway. If ADSRX or DICBRX are zero or are not input, then standard default values of 10s and 2.0 nmi are used. The default values will be overwritten if the user then enters proper values for ADSR (line 12) and DICBR (line 13). Of course, ADSR and DICBR will only be used when required.

5.5 PROB Changes

In the course of modifications to subroutine TWOPA (two parallel runways), changes were also made to the utility subroutine PROB, which deals with normal distributions. Given $\Phi(z)$, the cumulative probability of the normal distribution, PROB will calculate the standardized variable z, where

$$z = \frac{x - \overline{x}}{\sigma} \tag{21}$$

PROB would also calculate $\Phi(z)$ given z, by calling subroutine CUMPRO.

Both original routines PROB and CUMPRO were replaced with polynomial approximations for greater speed and accuracy. These were obtained from the IBM Scientific Subroutine Package (Reference 7), and are based upon equations published by Hastings (Reference 8) and Abramowitz and Stegun (Reference 9). These equations are:

$$\text{A)} \quad \Phi(z) = 1 - D(z)(b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4 + b_5 t^5)^* \tag{22}$$

$$\begin{array}{rcl} \text{where } D(z) & = & 0.3989423\ e^{-z^2/2} \\ t & = & 1/(1 + 0.2316419 * |z|\ ) \\ b_1 & = & 0.31938\ 1530 \\ b_2 & = & -0.35656\ 3782 \\ b_3 & = & 1.78147\ 7937 \\ b_4 & = & -1.82125\ 5978 \\ b_5 & = & 1.33027\ 4429. \end{array}$$

* Reference 9, Eq. 26.2.17 and Reference 7, subroutine NDTR.

The maximum error of this approximation is less than 7.5×10^{-8}.

$$\text{B)} \quad z = t - \frac{c_0 + c_1 t + c_2 t^2}{1 + d_1 t^2 + d_2 t^3 + d_3 t} \quad * \qquad (23)$$

where $t = \sqrt{\ln(\Phi^{-2}(x))}$

$c_0 = 2.515517$
$c_1 = 0.802853$
$c_2 = 0.010328$

$d_1 = 1.432788$
$d_2 = 0.189269$
$d_3 = 0.001308$

The maximum error here is 4.5×10^{-4}.

As an example of the improvement in accuracy obtained by use of these approximations, for a value of -2.927, the original program gave a $\Phi(z)$ of .001877, and the revised program .001711. The handbook value is .001712.

Use of these approximations has also reduced the running time of the program slightly. This is particularly true in subroutine DUAL, where the equation for $\Phi(z)$ has been added directly to the subroutine: the overhead of calling PROB separately has been eliminated. Also, CUMPRO has been eliminated as a separate subroutine, since the logic is all within PROB.

5.6 Changes to CRAIG, the Interactive Input Subroutine

The model user has a choice of two methods for inputting data to the capacity program:

- o create a batch input file from scratch or by editting a previously existing file, or
- o use the interactive capability of the program, which creates an input file based on the user's responses to questions.

* Reference 9, Eq. 26.2.23 and Reference 7, subroutine NDTRI.

Many changes have been made to CRAIG, the interactive subroutine. Some changes to CRAIG were required by other program changes (adding the special alternating arrival inputs, for example). Which questions were asked, and how those questions read, were also changed.

Other changes implemented the guidelines contained in FAA-EM-78-8A, "Parameters of Future ATC Systems Relating to Airport Capacity/Delay" (Reference 10). CRAIG now contains four standard ATC scenarios; these are explained and the associated characteristics are listed in Table 5-1.

The logic in CRAIG for deriving ADSR and DICBR values has also been modified. The new model 6-1 (intersecting: D,D) requires the time to clear the intersection, for each aircraft class, as input. CRAIG requests threshold-to-intersection distances for each runway; these distances are used to calculate the departure intersection-clearance time, the required departure/arrival separations (DICBR), and the arrival intersection-clearance time (ADSR). Previously, ADSR and DICBR were derived from tables.

For departures, the following assumptions were made:

- o Aircraft accelerate at a constant rate until liftoff, then fly at a constant speed.
- o Acceleration rate is 6 ft/s^2 for all aircraft classes.
- o Liftoff occurs at 1.4 Vs, where Vs is the stall speed. The approach speed is assumed to be 1.3 Vs.

The time to the intersection is then obtained by simple application of the equations of motion. The results are graphed in Figure 5-6.

Knowing this time to the intersection (which we will call TXI(k,nl), for departure k and runway nl) and knowing that in IMC the arrival must be 2.0 nmi from the intersection at that point, we can calculate DICBR:

$$DICBR(i,k) = (2.0 - d) + V(i) * TXI(k,nl) \qquad (24)$$

where d = the distance from arrival threshold to intersection

V(i) = the approach speed of arrival i.

In VMC, the arrival can be at the threshold, and

$$DICBR(i,k) = V(i) * TXI(k,nl). \qquad (25)$$

TABLE 5-1

EXPLANATION OF ATC SYSTEM CODES

THE FOLLOWING ATC SCENARIOS REPRESENT FAA E & D PLANNING AS OF JANUARY, 1980, AS DESCRIBED IN FAA-EM-78-8A.

ATC CODE	TIME FRAME	DESCRIPTION
P	PRESENT	CURRENT ATC SYSTEM
N	NEAR-TERM	VAS, TERMINAL FLOW MANAGEMENT
I	INTERMEDIATE	WVAS, TERMINAL FLOW MANAGEMENT, REDUCED RUNWAY OCCUPANCY IN IMC
F	FAR-TERM	WVAS, ADVANCED TERMINAL FLOW MANAGEMENT FURTHER REDUCTIONS IN IMC RUNWAY OCCUPANCY

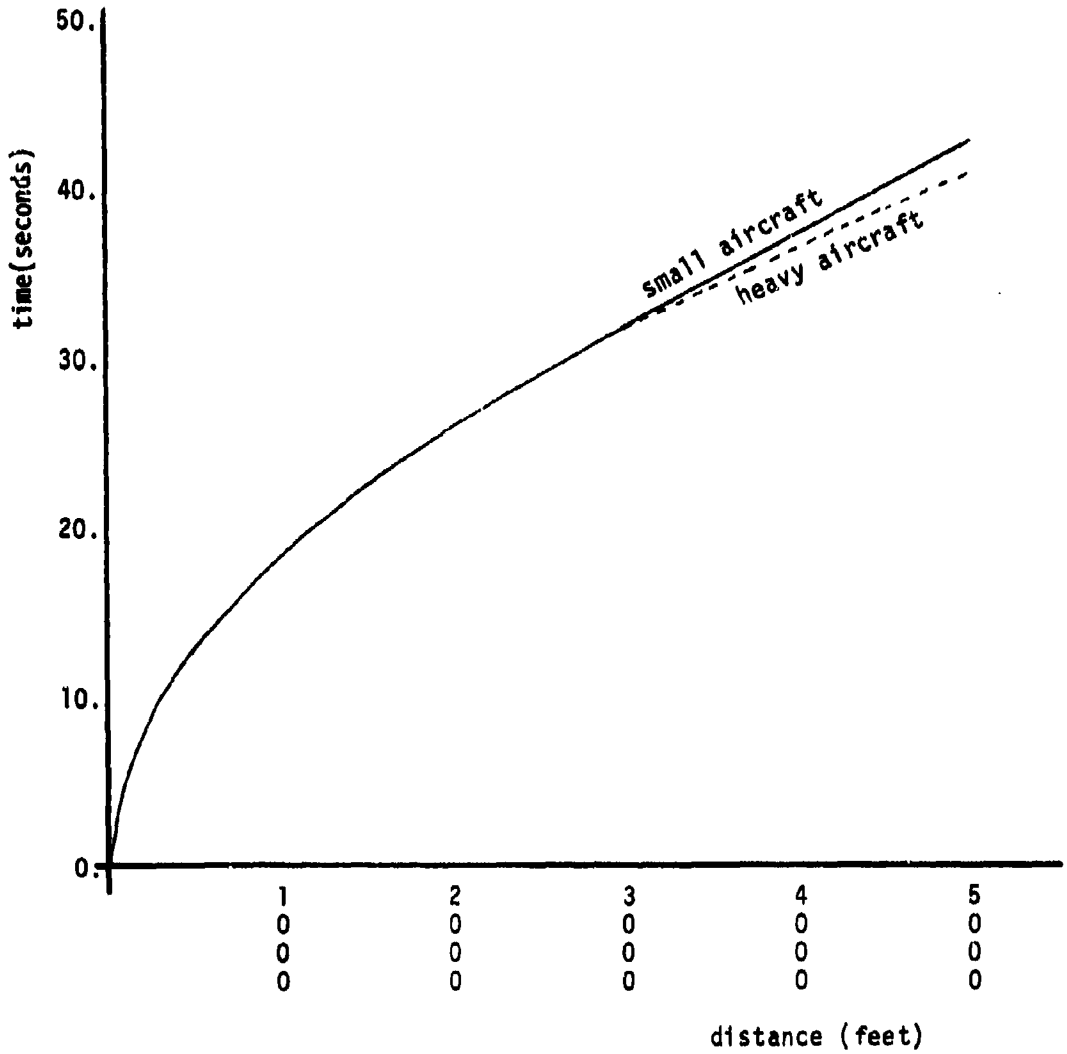


FIGURE 5-6
DEPARTURE INTERSECTION CLEARANCE TIME

Figure 5-7 compares the values of DICBR obtained from the original and revised logic.

The calculations for arrival times to intersection are slightly more complex. This is because the arrival's velocity profile is broken into four separate phases:

- o Threshold to touchdown
  - -- Speed at touchdown is assumed to be .95 V(i)
  - -- Touchdown is assumed 1500 feet down the runway
- o Touchdown to taxi speed
  - -- Arrivals decelerate at constant 5.3 ft/s^2 to the runway taxi speed of 60 knots
- o Taxi down runway
  - -- Constant speed of 60 knots is assumed
- o Runway taxi to exit speed
  - -- Short distance before exit, arrivals decelerate to exit speed (not considered here - arrivals either exit before the intersection, or roll through at runway taxi speed).

These assumptions are based in part upon the information in Reference 11. Figure 5-8 compares the original table of ADSR values with the new values, computed according to the above assumptions.

Normally, ADSR is simply the time for the arrival to clear the intersection. If both the arrival and intersecting departure are airborne at the runway crossing, however, ADSR must account for the required vortex separation at the crossing (120s for any aircraft following a heavy, presently). In this case,

$$ADSR(i,k) = \max\,[TXIA(i),\ TXIA(i) + ADV - TXI(k,\ n1)] \qquad (26)$$

where TXIA(i) = the time to intersection of arrival i
ADV = the arrival-departure time due to vortex (120s)
TXI(k,n1) = the time to intersection for the departure.

Similarly, DICBR must consider this vortex separation at the airborne intersection, as follows:

$$DICBR(i,k) = V(i)^* \max[TXI(k,n1),\ TXI(k,n1) + ADV - TXIA(i)]. \qquad (27)$$

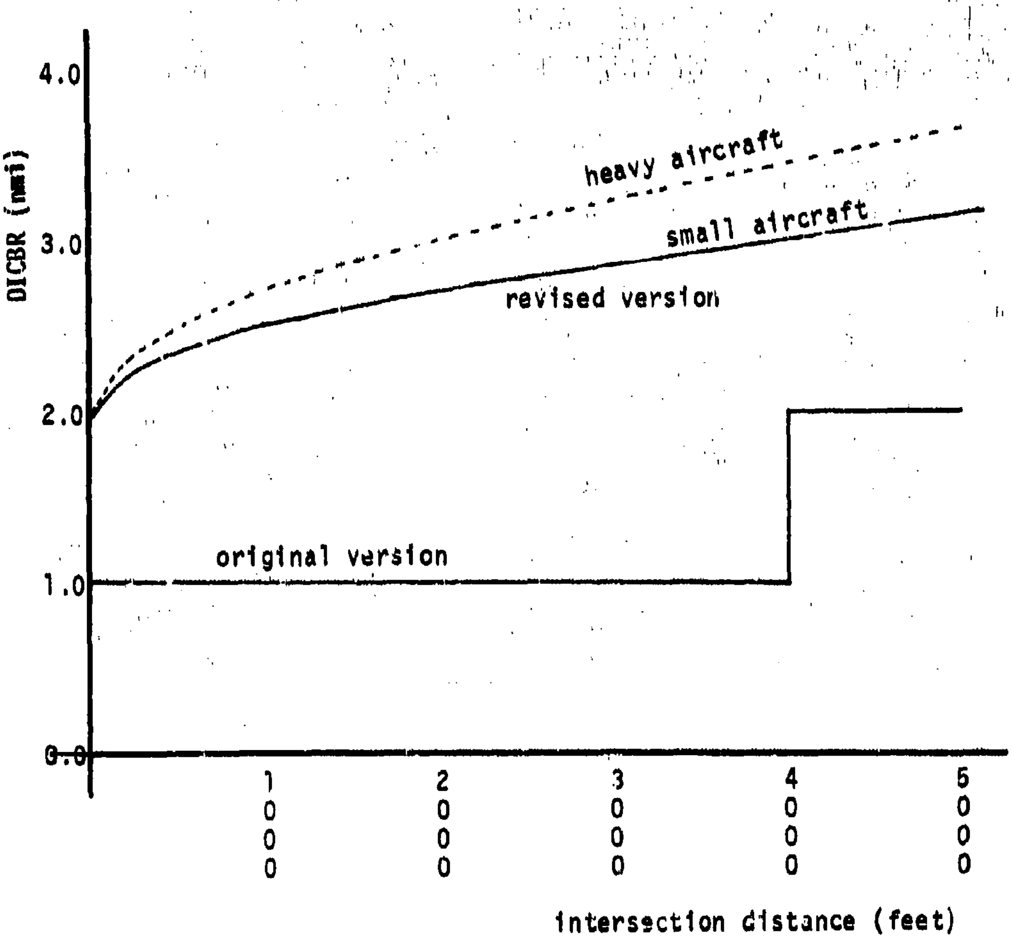


FIGURE 5-7
ORIGINAL AND REVISED VALUES OF DICBR

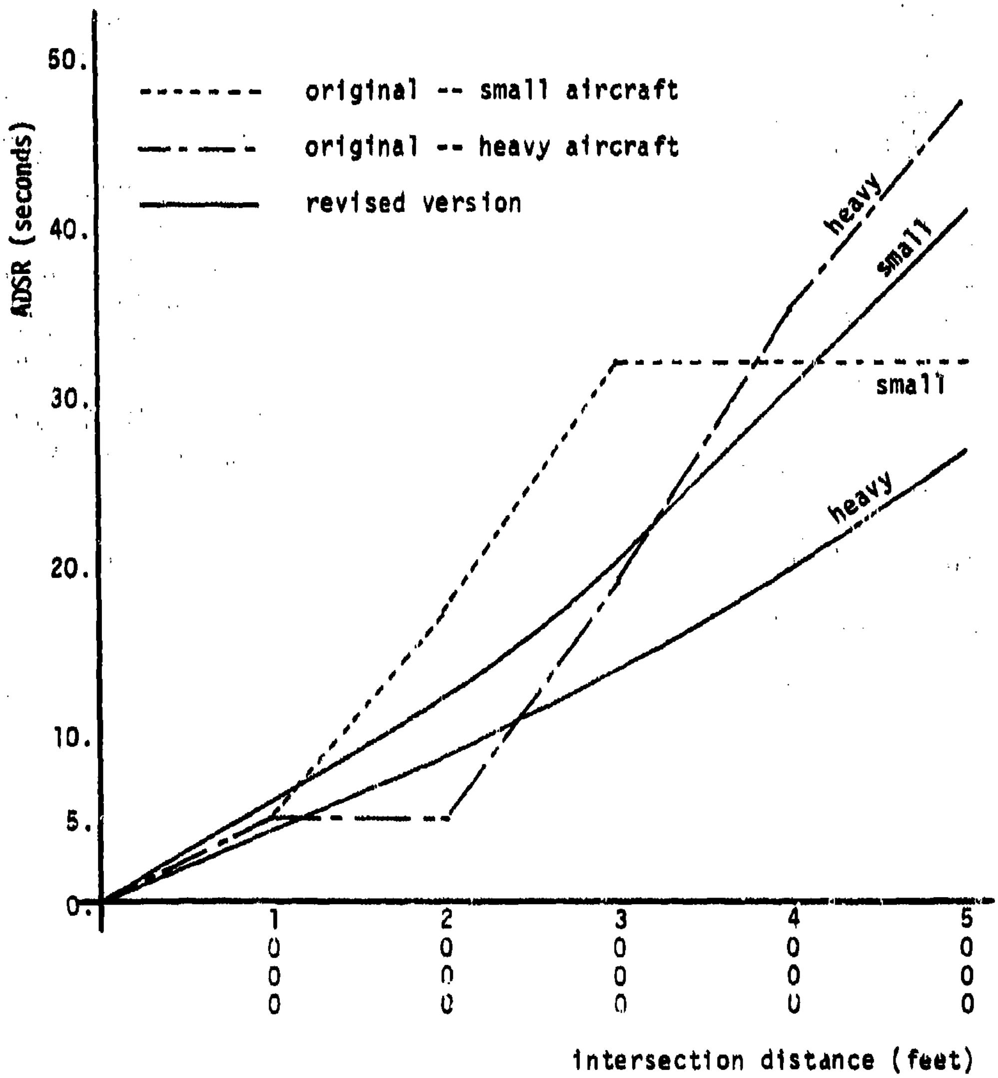


FIGURE 5-8
ORIGINAL AND REVISED VALUES OF ADSR

6. SUMMARY

This report has described the recent modifications and additions to the FAA's airport capacity program. A careful review of the original program revealed areas where changes were needed to bring the program up-to-date, to add worthwhile new capabilities, and to correct logic errors.

Among the changes which have been described are these:

- Use of selective stretching of arrival gaps to increase departure capacity
- Consideration of the "first enqueued departure" mix as distinct from the overall fleet mix
- The ability to specify more than one arrival percentage in a given run
- Calculation of capacity for alternating arrivals to parallel runways
- Adjustment of the decomposition of complex configurations into one or more simpler configurations.

Details of the modifications have been described, comparisons have been made between the original and the revised versions, and in some cases, the reasons for not implementing a proposed modification have been explained.

The program modifications described in this report, extensive as they may seem, are only a portion of all the revisions which were made to the capacity program. Other changes were made to improve the program running time, decrease the storage requirements, and increase the usability of the program and the accuracy of the result. Some such changes were:

- Common blocks were restructured so that fewer variables were stored in COMMON, and fewer variables were passed unnecessarily between subroutines
- Separate subroutines for calculating gate and taxiway capacity, rarely used, were eliminated
- DO loops were modified for greater efficiency.

In addition, many comment statements were added, and the program input and output were modified.

The result of all these modifications is a greatly improved program, easier for the first-time user to deal with but also with more options available to the experienced user, which will not (we hope) produce self-contradicting or unreliable results.

The overall effect of the changes which have been made may be judged by comparing the results obtained from both the original and revised versions of the program. One such comparison is made in Table 6-1. VMC and IMC capacities were calculated for Miami International Airport, using input data from Reference 5, the Airport Capacity Task Force report. The revised program calculated a capacity which was 11% higher in VMC, and 2% higher in IMC, than the original program. As can be seen from the capacity curves of these two cases (Figure 6-1), most of the increase comes not from a higher arrivals-priority capacity, but from the use of intermediate capacity points. It is also worth noting that the results from the revised program show a relative difference between VMC and IMC capacities that is in closer agreement with current-day experience than the results of the original version.

It must be realized, though, that this model will never be perfect. Changes to ATC procedures, or to predictions of future ATC procedures, will require appropriate updating of the program. New features and capabilities will be desired. Experience with the revised program may reveal previously hidden errors which need to be corrected, or areas of the program where the logic could be improved further. Among the areas where further improvement is possible are the logic for including touch-and-go operations, new logic for model 6-3 (intersecting: B,D), and logic for three intersecting runways (subroutine THREIN) which does not ignore the crossing runway.

A periodic general review and updating of the entire program should therefore occur. It need not be as extensive as this review and upgrading have been, provided that excessive time does not elapse between reviews.

TABLE 6-1

COMPARISON OF MIA RESULTS, ORIGINAL AND REVISED PROGRAM VERSIONS

		ORIGINAL VERSION	REVISED VERSION	CHANGE
VMC	CAPACITY	100.9	112.3	+11.3%
	COST	12.7 CPUs	14.5 CPUs	+14.2%
IMC	CAPACITY	99.9	102.2	+ 2.2%
	COST	14.2 CPUs	14.5 CPUs	+ 2.1%

-- MIAMI TODAY

-- FAR-SPACED PARALLELS, MIXED OPERATIONS ON BOTH

-- CAPACITY AT 50% ARRIVALS

-- 1 CPU second = $.20 (MITRE IBM 370/148)

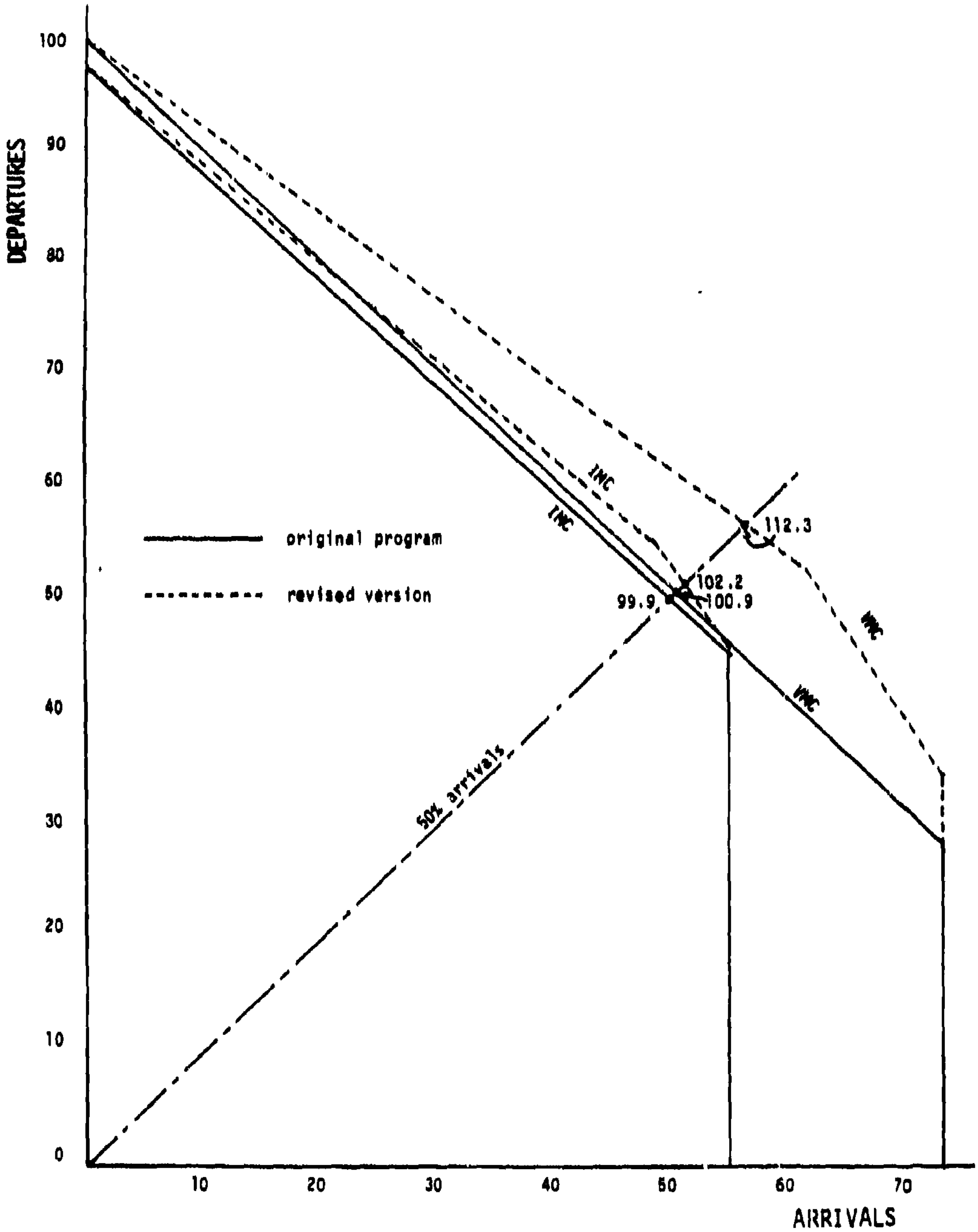


FIGURE 6-1
MIAMI CAPACITY CURVES, ORIGINAL AND REVISED PROGRAM VERSIONS

APPENDIX A

GLOSSARY

A	-	arrivals.
A	-	first of four classes of aircraft (ABCD), usually small.
A	-	the time period during which departure n would have an effect on departure k.
AA	-	arrival/arrival.
AD	-	arrival/departure.
ADSR(i,j)	-	arrival/departure separation requirement -- for intersecting runways, the minimum time after arrival i crosses the threshold at which departure j can be released.
ADSRX	-	a single value of ADSR which applies to all values of i and j.
ADV	-	minimum arrival/departure time at an intersection due to vortex restrictions.
ALTARR	-	line 25 of the input file, containing information needed for alternating arrival operations.
ALTTAA(n,i,j)	-	the average interarrival time between arrival i on runway n and arrival j on the parallel runway (alternating arrivals).
ARBAR(i)	-	the average runway occupancy time of aircraft i.
AROR1(i)	-	protected runway occupancy time of aircraft i -- ARBAR(i) plus a buffer.
ATC	-	air traffic control.
A1	-	the arrival capacity under conditions of arrival-priority.
A2	-	the arrival capacity under conditions of departure-priority.
B	-	second of four aircraft classes (ABCD), usually either small or large.
B	-	both arrivals and departures on the same runway -- mixed operations.
C	-	close-spaced parallel runways (700-2499 feet apart).
C	-	third of four aircraft classes (ABCD), usually large.
Cap	-	capacity.
CLDIST	-	distance between centerlines of two parallel runways.
CNV	-	covergence criterion used in f.e.d. mix calculations.
CONDAB	-	program subroutine which determines degree of dependence between two non-parallel, non-intersecting runways.
CPU	-	central processing unit -- used as a measure of program execution time (also CPU second).

CRAIG	- program subroutine which constructs an input file based upon user responses to a series of questions.
CUMPRO	- program subroutine which formerly calculated (z) given z -- now included as part of subroutine PROB.
d	- distance between thresholds of non-parallel, non-intersecting runways.
D	- last of four aircraft classes (ABCD), usually heavy.
DA	- departure/arrival
DASR(j)	- departure/arrival separation requirement -- the time required for arrival j to fly DLTADA, plus a buffer.
DD	- departure/departure.
DDSR	- departure/departure separation requirement -- minimum time between departures, in seconds.
DELTA	- the incremental time (in seconds) by which arrival gaps are stretched.
DIAGSP	- the diagonal separation required between alternating arrivals on separate runways.
DICBR(i,j)	- on intersecting runways, the minimum distance arrival j can be from the threshold when departure i is released.
DICBRX	- a single value of DICBR which applies to all values of i and j.
D(k)	- the expected number of departures of type k in a single arrival gap.
DLTADA	- the minimum distance an arrival must be from the threshold in order to release a departure on the same or a close-parallel runway.
DLTAIJ(i,j)	- the minimum airborne separation required between lead aircraft i and trail aircraft j.
DROR2(k)	- departure runway occupancy requirement of aircraft type k -- average occupancy plus a buffer.
DUAL	- program subroutine for calculating the capacity of a dual-lane runway (IMC only).
D1	- the departure capacity under conditions of arrival-priority.
D2	- the departure capacity under conditions of departure-priority.
e	- the base of the natural logarithms, approximately 2.71828.
EPSILN	- the distance at which an arrival can first see the runway end - the minimum of visibility and ceiling/tan GS, where GS is the angle of the glide slope.
Exp	- exponential. $Exp[x] = e^x$.
E1	- an element in the expression for the convolved probability P1*, equal to $Exp[-(DDSR(n,k) - \alpha^2/2 * SIGMAA^2]$.
E2	- an element in the expression for the convolved probability P1*, equal to $Exp[-\beta^2/2 * SIGMAA^2]$.

F - far-spaced parallel runways (more than 4300 feet apart).
FAA - Federal Aviation Administration.
f.e.d - first enqueued departure -- refers to the probability that a particular aircraft type will be the first in line to depart, as different from the overall proportion of that type in the fleet.
FINC - the increment used for DLTAIJ in subroutine SUPER.

g - the arrival aircraft preceding the current pair ij.
GMAX - the maximum value of DLTAIJ used in subroutine SUPER.
GTDISP - the relative displacement of the final approach gates, used for alternating arrivals.

H - heavy aircraft - maximum gross takeoff weight of 300,000 pounds or more.
HH - heavy following heavy.
HS - a small following heavy.

i - the lead aircraft in the current arrival pair ij.
IALT - flag which indicates to subroutine TWOPA whether or not alternating arrivals are to be modeled.
IAT - interarrival time -- time between successive threshold crossings.
IAX - flag used to indicate whether aircraft are airborne at the runway intersection.
IBOMB - variable used to count the number of iterations through the f.e.d. mix logic.
IFR - Instrument Flight Rules.
ijk - a scalar representing the combination of arrivals i and j and departure k.
IMC - Instrument Meteorological Conditions.
IMODEL - the model series of the configuration being analyzed.
IR - flag returned by subroutine CONDAB to indicate the degree of dependence between two non-parallel, non-intersecting runways.
ISTRGY - the original operating strategy of the configuration being analyzed.

j - the trail aircraft in the current arrival pair ij.
JBOMB - the maximum number of iterations to be performed by the f.e.d. mix logic.
JST - a variable used to count the number of intermediate points whose capacity has been calculated.

k - the first departure in the ij gap.
KMODEL - the model series of the configuration which is the departure-priority equivalent of the configuration being analyzed.
KSTRGY - the operating strategy of the equivalent departure-priority configuration.

l	- the second departure in the ij gap.
L	- large aircraft - maximum gross takeoff weight of 12,500 pounds or more, but less than 300,000 pounds.
LGA	- LaGuardia Airport.
LH	- a heavy following a large.
LL	- a large following a large.
LS	- a small following a large.
m	- the third departure in the ij gap.
M	- medium-spaced parallel runways - originally 3500-4299 feet apart, but currently 2500-4299 feet apart.
max	- maximum.
MIA	- Miami International Airport.
min	- minimum.
MIXOP	- program subroutine which calculates the mixed operation capacity of a single runway.
MMC	- Marginal Meteorological Conditions.
M1	- in the Q-logic, the probability that departure k in the current gap is not affected by departures in the previous gap, considering all possible values of n.
n	- the type of the last departure in the previous gap, between arrivals g and i.
n'	- those values of n such that DDSR(n,k) > TND(i).
N	- near-spaced parallel runways (2500-3499 feet apart) -- rarely-used category today.
NEWRUN	- line 0 of the input file, containing values of IMODEL, ISTRGY, and ISTAG.
NST	- the maximum number of intermediate points whose capacity is to be calculated.
OPENV	- line 10 of the input file, containing information needed for non-parallel, non-intersecting runway configurations.
P	- probability. P[x] = probability of event x.
PFED(i,k)	- the probability that departure k is the first enqueued departure after arrival i.
PHR(k)	- the proportion of type k in the overall fleet mix.
PMM&Co.	- Peat, Marwick, Mitchell and Company.
PNE	- in the derivation of the convolved probability P1*, the probability that departure n can never affect departure k.
PROB	- program subroutine which calculates $\Phi(z)$ given z, or z given $\Phi(z)$.
PST	- in the logic for selective gap stretching, the required probability that another departure can be released in the stretched gap.

PTA(ni,j) - the array used to store the capacity values at intermediate points -- the arrival capacity of point ni is PTA(ni,1), and the departure capacity is PTA(ni,2).

PTEMP(i,k) - array used for temporary storage of the f.e.d. mix of departure k following arrival i during the calculation of the final f.e.d. mix.

PT1(n,ijk) - an element of the expression for the convolved probability P1*, equal to $\frac{\text{DDSR}(n,k) - a}{\text{SIGMAA}}$.

P1A(n,ijk) - the probability that departure k can be released in the ij gap, considering the effect of previous departure n.

P1A1(n,ijk) - the probability that n occurs too early to affect k and the ij gap is large enough to release k.

P1A2(n,ijk) - the probability that n does affect k, but the ij gap is large enough to release k anyway.

P1(ijk) - the probability of releasing departure k (and possibly others) in the ij gap.

P1*(ijk) - the revised probability of releasing k in the ij gap, considering the possible effects of previous departures.

P2(ijk,l) - the probability of releasing k and l in the ij gap.

P2*(ijk,l) - the revised probability of releasing k and l in the ij gap, considering the possible effects of previous departures.

P3(ijk,l,m) - the probability of releasing k,l, and m in the ij gap.

P3*(ijk,l,m) - the revised probability of releasing k,l, and m in the ij gap, considering the possible effects of previous departures.

Q-logic - the program logic by which the possible effects of previous departures are accounted for.

Q(n,i) - the probability that n is the last departure before arrival i.

S - small aircraft, maximum gross takeoff weight of less than 12,500 pounds.

S• - a small aircraft followed by any type.

SIGAI - the standard deviation of the time for an arrival to clear the runway intersection.

SIGALT(n,i,j) - the standard deviation of the time between alternating arrivals i on runway n and j on the parallel runway.

SIGBRS(n,i,j) - the standard deviation of the time between alternating arrivals i and j, both on runway n.

SIGDI - the standard deviation of the time for a departure to clear the runway intersection.

SIGMAA - the standard deviation of the interarrival time.

SIGMAR - the standard deviation of the arrival runway occupancy time.

SINGLE	- program subroutine for calculating the basic capacity of a single runway.
SLACK	- a variable in the original Q-logic, equal to $\overline{TAA}(i,j)$ + DLTADA/V(i) - DLTADA/V(j).
SRDS	- Systems Research and Development Service.
STAGGR	- program subroutine which calculates capacity of alternating arrivals to parallel runways.
SUPER	- program subroutine which provides a crude form of capacity maximization.
S5	- weighted average of time between departures #3 and #5 in logic for intersecting departures.
t_i	- the time at which arrival i crosses the threshold.
t_j	- the time at which arrival j crosses the threshold.
t_k	- the time at which departure k is released.
t_n	- the time at which departure n was released.
Tn	- time at which the nth departure is released, in the logic for intersecting departures.
TAABRS(n,i,j)	- the average interarrival time between alternating arrivals i and j, both on runway n.
$\overline{TAA}(i,j)$	- the average interarrival time between arrivals i and j.
TAA(i,j)	- the actual interarrival time between arrivals i and j.
TAASAV(i,j)	- in selective incremental gap stretching, the largest value of interarrival time for which a benefit has been calculated, against which TAATST will be compared.
TAATST(i,j)	- the test value of interarrival time.
TGTi	- the time at the final approach gate of the ith aircraft (alternating arrivals).
TH	- runway threshold.
THDISP	- for alternating arrivals, the relative displacement between runway thresholds.
THREIN	- the program subroutine which calculates the capacity of three intersecting runways, two of which are parallel.
TND(i)	- the time prior to arrival i during which no departure can be released, equal to (DLTADA - EPSILN)/V(i).
TTHi	- the time at the threshold of the ith aircraft (alternating arrivals).
TTHji	- the time at the threshold for arrival j, as determined solely by the required separations behind arrival i, where TGTi and TTHi are known (alternating arrivals).
TWOIN	- program subroutine which calculates the capacity of two intersecting runways.
TWOPA	- program subroutine which calculates the capacity of two parallel runways.
TXIA(i)	- the time from threshold to clearing the runway intersection for arrival i.
TXI(k,m)	- the time from release to clearing the intersection for departure k on runway m.

V(i)	- the final approach velocity of arrival i.
VFR	- Visual Flight Rules.
VMC	- Visual Meteorological Conditions.
XSEP	- the diagonal separation between alternating arrivals measured along the final approach path.
XST	- in the logic for selective gap stretching, the proportion of the unstretched gap to the gap size needed to release a departure with probability PST.
z	- the standardized normally-distributed variable, equal to $(x-\bar{x})/\sigma$.
α	- an element of the expression for the convolved probability P1*, equal to TAA(i,j) + DLTADA/V(i) - DLTADA/V(j).
α'	- an element of the expression for the convolved probability P2*, equal to α - DDSR(k,1).
α''	- an element of the expression for the convolved probability P3*, equal to α - DDSR(k,1) - DDSR(1,m).
β	- an element of the expression for the convolved probability P1*, equal to TAA(i,j) + EPSILN/V(i) - DLTADA/V(j).
β'	- an element of the expression for the convolved probability P2*, equal to β - DDSR(k,1).
β''	- an element of the expression for the convolved probability P3*, equal to β - DDSR(k,1) - DDSR(1,m).
θ	- the angle between two non-parallel, non-intersecting runways.
π	- the probability that departure k in the current gap is not affected by previous departure n.
$\sum_n$	- mathematical symbol indicating a sum over all values of n.
σ	- standard deviation
Φ	- the cumulative distribution function operation for a normally distributed variable.

APPENDIX B

GAP STRETCHING ALTERNATIVES

by Frank A. Amodeo

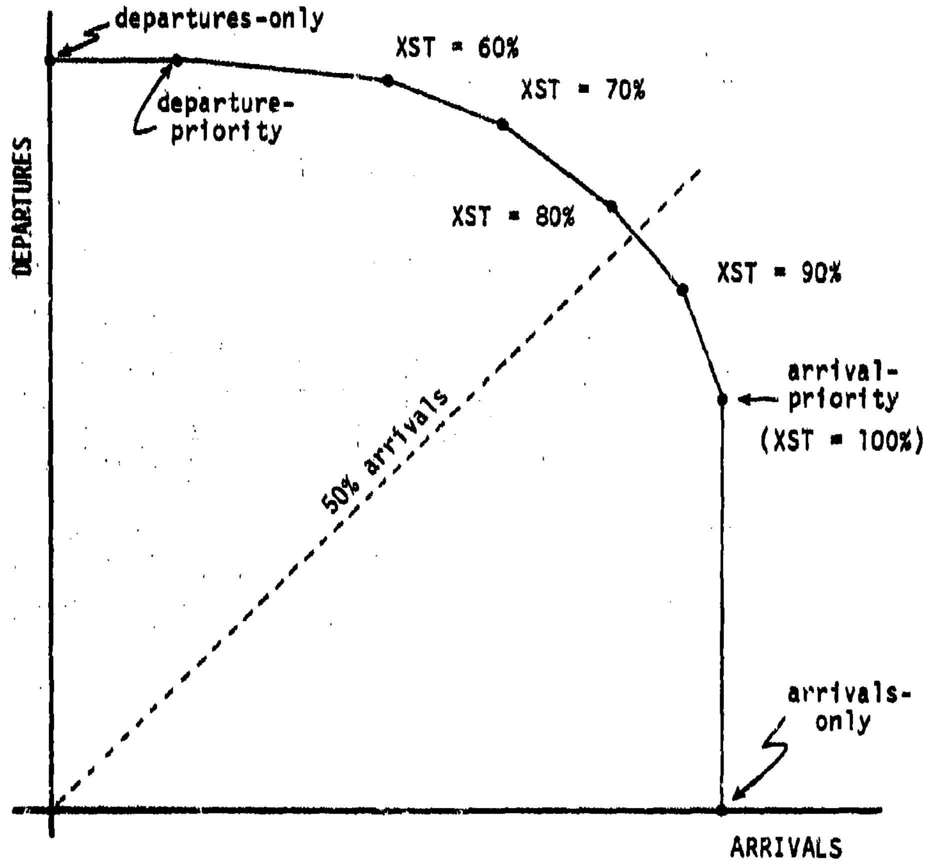
Described in this appendix are several alternative methods of stretching selected interarrival gaps to obtain capacity estimates over a range of arrival percentages. The three different algorithms described exhibit varying levels of accuracy, with the first version not working very well and subsequent versions showing improvements.

An assumption common to all the algorithms described in this appendix (but not to the algorithm presented in section 2.3) is that although the departures are random, the particular outcome of the random experiment, i.e., the departure queue, is known at the time that the arrivals are sequenced. That is, the interarrival spacing is stretched for particular classes of departures. A maximum of three departures per arrival gap is considered.

In the first attempt to develop gap-stretching logic, the following algorithm was proposed. Given interarrival gap i-j and departure queue k-l-m, if the size of the average (unstretched) interarrival gap is XST (an input variable, $0 \leq XST \leq 1$) of the size needed to get another departure (k, l, or m) out with probability PST, then the gap will be stretched by this amount. The parameter PST may be set to some suitable value, e.g., 0.99, since we want to ensure that the departure gets out with fairly high probability.

The variable XST is varied from 1.00, corresponding to the unstretched case, down to some lower limit. As XST is decreased, more (i,j,k,l,m) combinations have their associated interarrival gaps stretched, and gaps are stretched to a greater extent. This has the effect of reducing the arrival capacity and increasing the departure capacity. As XST is varied, different points on the capacity curve are calculated. An example of this is illustrated in Figure B-1, which shows the type of curve which the algorithm was expected to provide.

When this logic was implemented and tested, some anomalous results were obtained. Evenly spaced values for XST yielded points on the capacity curve which were clustered. This was not really unexpected because there is no reason to think that evenly spaced values for XST would yield uniformity in the number of combinations (i,j,k,l,m) of arrivals and departures for which the interarrival gap is stretched or the amount by which these gaps are stretched.



NOTE: Values of XST are shown for illustration only

FIGURE B-1
CAPACITY CURVE CONCEPT

More importantly, the curves yielded by this algorithm did not possess some of the theoretical properties which were postulated, such as concavity. That is, as we decrease XST, the resulting capacity point should lie on or above the line connecting the departures-only capacity and the capacity point due to the previous value of XST, and be on or below the line passing through the previous two points. The points generated by this algorithm indicated that several problems existed with this logic. Particular points lay above the extrapolation line defined by the previous two points. Ultimately, the curve turned downward, below the line between a previous point and the departures-only point, and in most cases the last point had a smaller departure capacity than the next-to-last point. This demonstrated that the logic used to determine which gaps to stretch and by how much was faulty and that a radical redesign was called for.

A subsequent algorithm attempted to address the problems exhibited by the first algorithm. A new criterion was developed, whereby the option of stretching a gap for an additional departure was compared to the alternative of operating in a departures-only mode part of the time.

In a departures-only case, the expected number of departures per second is equal to the inverse of the average DD spacing. In the case of mixed operations, the expected number of departures per gap is the sum of three cumulative normal distributions. An example is shown in Figure B-2. The slope of this function is the expected number of additional departures per additional second of interarrival time. This slope is the sum of the three normal density functions and is therefore available in closed form.

If, for a particular combination of (i,j,k,l,m), we initially are on the curve (see Figure B-2) at a point with a slope greater than the reference slope, say point A, then it is worthwhile to stretch gap i-j, relative to running departures-only for part of the time. As we add incremental units of time to the gap, it remains efficient until the slope is reduced to the reference slope; this occurs at point B. Further stretching beyond point B is inefficient, and in terms of the type of curve depicted in Figure B-1 would yield a point below the interpolation line between the last calculated point and the departures-only point. If the slope at the starting point is less than the reference slope (point C), then it would be more efficient to run a departures-only configuration part of the time to obtain the desired percent arrivals. We see that some of the problems with the earlier version may have been due to the fact that if we stretched a gap for a particular departure, we wanted to make sure that the departure was released with fairly high probability. Setting this probability (PST) too high results in stretching to a point on the curve shown in Figure B-2 where the curve is fairly flat. However, stretching to a point on the curve which represents a lower probability of successful departure

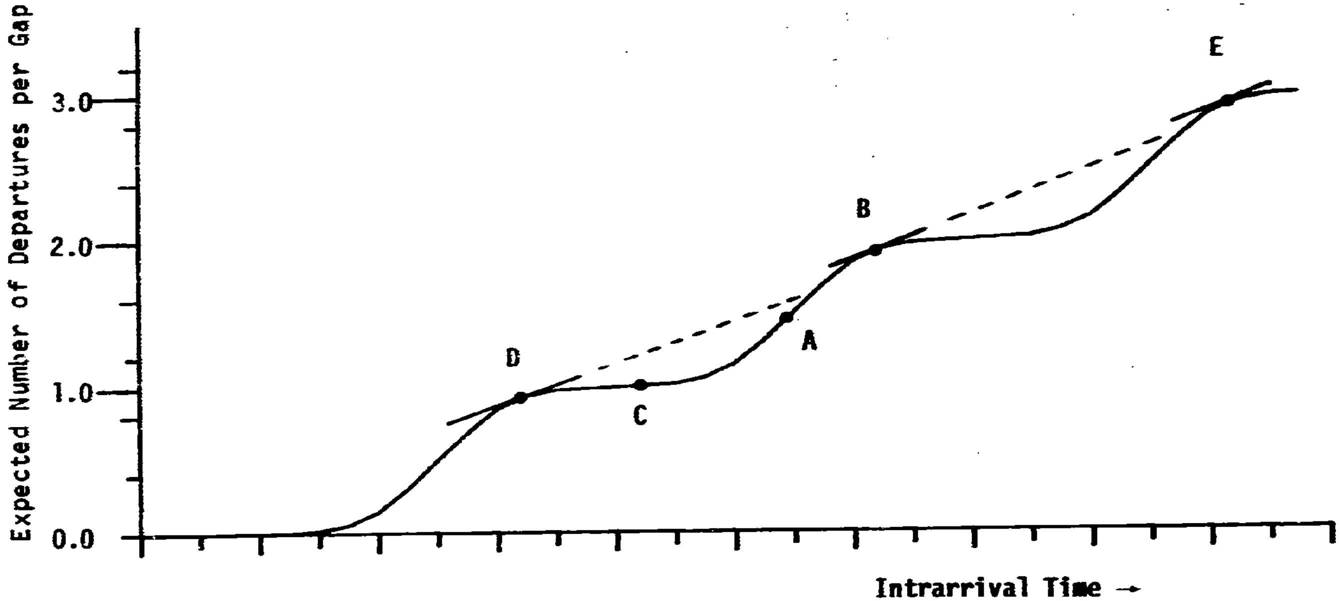


FIGURE B-2
CURVE OF DEPARTURES PER GAP vs. SIZE OF GAP

release amplifies the question of what effect missing a departure has on capacity, when gaps are stretched for particular departures. At the expense of leaving this question unresolved, the revised algorithm would stretch a gap to one of these "optimal" points (such as point B, D or E in Figure B-2). The revised algorithm is limited to stretching only those gaps whose unstretched points lie on a steep part of the curve (Figure B-2) such as point A. This does not account for all available efficiency since the algorithm does not stretch a point such as C up to point B (Figure B-2), even though the value of ΔE [departures]/ ΔAA(i,j) would be greater than the reference slope. The algorithm does not stretch any gap more than once. Some efficiency may be lost here since to move from "optimal" point D to "optimal" point B in Figure B-2 would yield a value of ΔE [departures]/ ΔAA(i,j) greater than the reference slope.

The algorithm orders all combinations of (i,j,k,l,m), for which gap stretching is determined to be efficient, by their values of ΔE [departures]/ ΔAA (i,j). It then stretches the gaps in order of decreasing benefit. The algorithm stretches for a number of combinations of (i,j,k,l,m) before calculating the capacity. The capacity is calculated when both of the following criteria are satisfied: 1) when the total probability of combinations (since the last capacity calculation) is greater than PMASS, an input variable; and 2) when all additional combinations having a benefit equal to that of the combination satisfying the first criterion are stretched. This second criterion was added to avoid unnecessary perturbation of the f.e.d. (first enqueued departure) mix (see section 3.1). Although the reference slope used in the algorithm is the inverse of the average DD spacing, an alternative which was proposed is the inverse of the particular DD corresponding to the pair in question. That is, if we can release departure k with fairly high probability and are trying to determine the efficiency of stretching the gap in order to release departure l, then the reference slope for this case would be $[DD(k,l)]^{-1}$.

While this question has not been resolved definitively, it was decided to use the average DD for the following reason. The departures that would be released in a departures-only mode, to make up for departures not released in interarrival gaps, would be random and in proportion to the aircraft mix. That is, a specific DD separation not added to an interarrival gap would be made up by adding an average DD separation during the departures-only mode. The fact that particular DDs appear to be more favorable than others, due to the fact they compare more favorably to the reference slope, impacts the capacity through the f.e.d. mix, but does not change the mix of excess departures released during a departures-only period.

The results from the second version exhibited the theoretical properties which were hypothesized. The resulting curves were concave and the problems with the first version, such as decreased departure capacity with decreased arrival capacity for the last one or two points, were not present. However, the capacity estimates for those points corresponding to stretched gaps were probably underestimates due to the fact that not all potentially efficient gaps were stretched, as described above.

A third version of this algorithm, while not addressing this problem, attempted to reduce the running time of the program. This algorithm stretches all of the gaps for which a benefit would occur and then calculates the capacity. This version gives a subset of the points (on a curve such as Figure B-1) yielded by the second version. These points are the arrivals-only capacity, departures-only capacity, arrival-priority capacity (unstretched) and the capacity point corresponding to all gaps being stretched (excluding those which are not efficient). The advantage of this third version is that fewer passes are made through the capacity-calculating routine and there is no need to sort the (i,j,k,l,m) combinations by their benefits. This third version also avoids a philosophical problem with the intermediate capacity points of the second version. This problem concerns stretching a gap to its "optimal" point. This point is only optimal relative to running in a departures-only mode part of the time. If other gaps are not fully stretched then there are three alternatives: stretching the gap in question, stretching another gap, and running departures-only part of the time. It may indeed be more efficient to only partially stretch one gap and then start to stretch another. Stretching all gaps fully before calculating the capacity, as is done in the third version, avoids this question.

At this point, further development of the sequence of algorithms using the assumption that interarrival gaps are stretched for particular classes of departures was abandoned. This was done because this assumption presented several problems, both practical and philosophical. The logic used to determine whether or not to stretch for a particular (i,j,k,l,m) and if so by how much required much computing time, due to the number of combinations (4^5, or 1024); furthermore, the indicated improvements would require still more computing time. The question of what effect missing a departure has on capacity when gaps are stretched for particular departures remained unanswered. It seemed optimistic to expect that information about the departure queue would be available far enough in advance to be used in sequencing arrivals. Finally, another algorithm which does not assume that arrivals are stretched for particular departures (described in section 2.3) was exhibiting very good results.

APPENDIX C

DERIVATION OF P1*, P2*, P3* BY CONVOLUTION OF DISTRIBUTIONS

This Appendix will present the derivation of the equations for P1*, P2*, and P3*, the probabilities of one or more, two or more, and three departures in a given arrival gap, accounting for the effect of departures in the previous arrival gap. The equations are derived for a dual lane runway, i.e., a close-spaced parallel pair with arrivals on one and departures on the other. IFR rules are assumed to apply: departures cannot be released if the arrival is closer than DLTADA (usually 2.0 nmi) to the runway threshold unless the arrival has the runway in sight (distance to threshold is EPSILN or less). Since it is on a separate runway, the departure is not constrained by the runway occupancy time of the arrival.

C.1 Derivation of P1*(ijk)

Figure C-1 uses a time-line diagram to depict the relevant interactions between two arrivals (i,j) and two departures (n,k). Some terms are also defined in this figure.

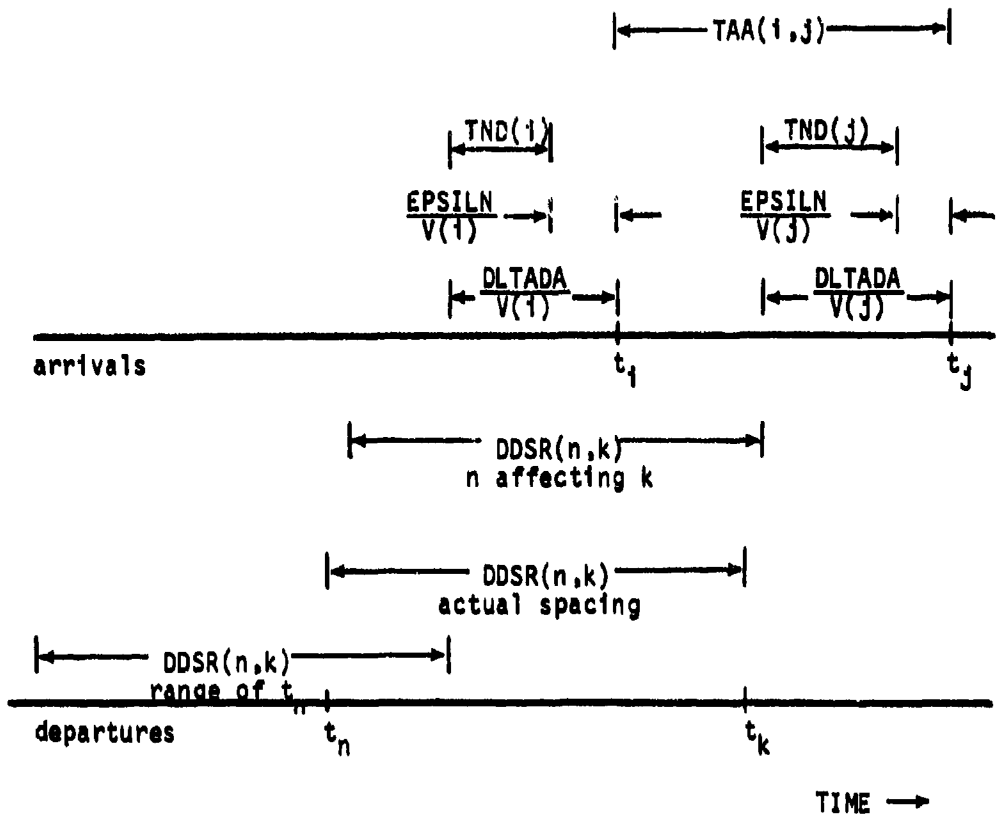
We shall first derive P1*(ijk), where we define

$$P1^*(ijk) = \sum_n P1A(n,ijk) * Q(n,i)$$

$$+ P1(ijk) * [1 - \sum_n Q(n,i)], \qquad (28)$$

P1A(n,ijk) is the probability that departure k can be released in the ij gap, considering the effect of departure n in the previous arrival gap. Q(n,i), which was derived in Section 3, is the probability that the last departure before arrival i was of type n. If there was no departure in the previous gap -- an event which has a probability of $1 - \sum_n Q(n,i)$ -- the probability of releasing departure K remains at P1(ijk).

Departure n can affect the departures in the current gap if the departure/departure separation requirement (termed DDSR(n,k)) causes the delay of departure k so that k can no longer be released in the ij gap. A certain minimum separation between departures is required by the departure/arrival and arrival/departure constraints (DLTADA and EPSILN, respectively). If DDSR(n,k) is less than or equal to this operational minimum, departure n cannot affect k. In other words,



t_i, t_j	time over threshold of arrivals i,j
t_n, t_k	time of release for departures n, k
TAA(i,j)	actual interarrival time
DLTADA	minimum distance of arrival from threshold in order to release a departure -- currently 2.0 nmi
EPSILN	distance at which visual contact with the runway is established
TND(i)	time during which departures may not be released
DDSR(n,k)	required departure-departure separation between n and k

FIGURE C-1
TIME AXIS DIAGRAM OF DUAL-LANE OPERATIONS FOR DERIVATION OF CONVOLVED PROBABILITIES

$$P1A(n,ijk) = P1(ijk) \quad \text{if } DDSR(n,k) \le (DLTADA - EPSILN)/V(i) \quad (29a)$$

where $V(i)$ = the velocity of arrival i.

The quantity (DLTADA - EPSILN)/V(i) will also be referred to as TND(i), the time during which no departures may be released.

If DDSR(n,k) is greater than the operational minimum time between departures, then it is possible for departure n to affect departure k, depending upon the time at which n occurs. Departure k may be released, in this case, if

- departure n occurs too early to affect k, and the ij gap is large enough to release k (the probability of which will be termed P1A1), or
- departure n causes k to be delayed, but the ij gap is large enough to accommodate the later release time for k (P1A2).

This can be expressed as

$$P1A(n,ijk) = P1A1(n,ijk) + P1A2(n,ijk) \quad \text{if } DDSR(n,k) > TND(i). \quad (29b)$$

C.1.1 Derivation of P1A1(n,ijk)

Departure n cannot affect k if n occurs more than DDSR(n,k) before the earliest time at which k can be released. If we call this probability PNE (see Figure C-1),

$$PNE = P\,[t_n < - EPSILN/V(i) - DDSR(n,k)] \quad (30a)$$

where t_n = the time at which n is released.

Note that t_i has been defined to occur at time zero.

We do not know the exact time at which n is released, since this is a function of previous arrivals and departures whose types are unknown. It will therefore be assumed that t_n is uniformly distributed over the feasible range. The upper limit on this range is -DLTADA/V(i), by definition of DLTADA; the lower limit is -DLTADA/V(i) - DDSR(n,k) because, if t_n were earlier, k could also be released in the gap, and n would no longer be the last departure in the previous gap. The range of t_n is therefore DDSR(n,k), and

$$PNE = \frac{(-EPSILN/V(i) - DDSR(n,k)) - (-DLTADA/V(i) - DDSR(n,k))}{DDSR(n,k)} \quad (30b)$$

$$= \frac{DLTADA/V(i) - EPSILN/V(i)}{DDSR(n,k)} = \frac{TND(i)}{DDSR(n,k)}. \tag{30c}$$

If n occurs too early to affect k, the probability of releasing k in the ij gap is simply P1(ijk). Therefore,

$$P1A1(n,ijk) = PNE * P1(ijk) = \frac{TND(i)}{DDSR(n,k)} * P1(ijk). \tag{31}$$

C.1.2 Derivation of P1A2(n,ijk)

The second component of P1A(n,ijk) is the joint probability that n occurs late enough to affect k, but the ij gap is nevertheless large enough to allow the departure of k. In other words,

$$\begin{aligned} P1A2(n,ijk) = P\ [&-EPSILN/V(i) - DDSR(n,k) \le t_n \le -DLTADA/V(i) \\ &\underline{\text{and}}\ t_j \ge t_n + DDSR(n,k) + DLTADA/V(j)] \end{aligned} \tag{32a}$$

where t_j = the time when j crosses the threshold.

This is equivalent to

$$P1A2(n,ijk) = \int_{-EPSILN/V(i) - DDSR(n,k)}^{-DLTADA/V(i)} g(t_n) \int_{t_n + DDSR(n,k) + DLTADA/V(j)}^{\infty} f(t_j)\,dt_j\,dt_n \tag{32b}$$

where $g(t_n)$ = the probability density function of t_n

$f(t_j)$ = the probability density function of t_j.

Since t_n is uniformly distributed, and t_j is normally distributed,

$$g(t_n) = 1/DDSR(n,k) \tag{33}$$

$$f(t_j) = \frac{1}{SIGMAA\sqrt{2\pi}}\, e^{\dfrac{-(t_j - TAA(i,j))^2}{2\ SIGMAA^2}} \tag{34}$$

The order and limits of integration can be changed by substitution (see Figure C-2) to yield:

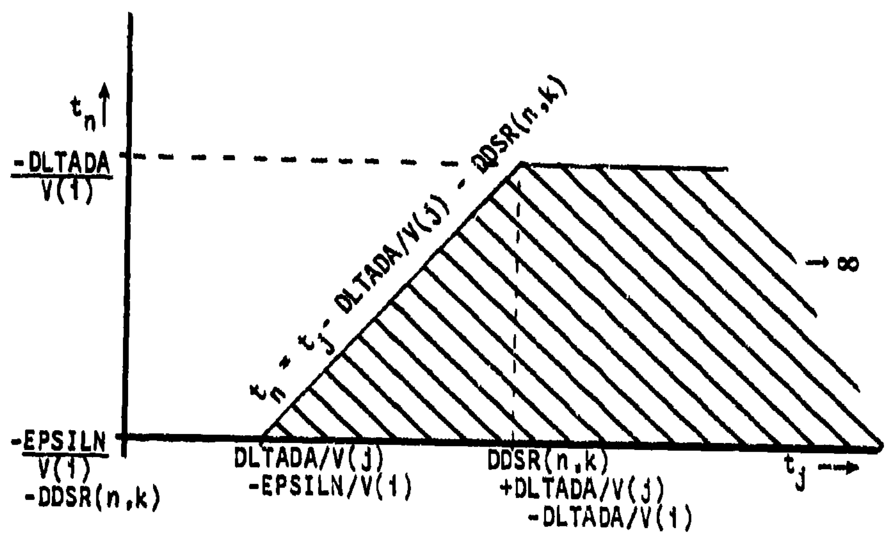


FIGURE C-2
LIMITS OF INTEGRATION

$$\mathrm{P1A2}(n,ijk) = \int_{\mathrm{DLTADA}/V(j) - \mathrm{EPSILN}/V(i)}^{\mathrm{DDSR}(n,k) + \mathrm{DLTADA}/V(j) - \mathrm{DLTADA}/V(i)} 1/\mathrm{DDSR}(n,k) \int_{-\mathrm{EPSILN}/V(i) - \mathrm{DDSR}(n,k)}^{t_j - \mathrm{DLTADA}/V(j) - \mathrm{DDSR}(n,k)} f(t_j)\, dt_n dt_j$$

$$+ \int_{\mathrm{DDSR}(n,k) + \mathrm{DLTADA}/V(j) - \mathrm{DLTADA}/V(i)}^{\infty} 1/\mathrm{DDSR}(n,k) \int_{-\mathrm{EPSILN}/V(i) - \mathrm{DDSR}(n,k)}^{-\mathrm{DLTADA}/V(i)} f(t_j) dt_n dt_j \qquad (35a)$$

$$\mathrm{P1A2}(n,ijk) = \int_{\overline{\mathrm{TAA}}(i,j) - \beta}^{\mathrm{DDSR}(n,k) + \overline{\mathrm{TAA}}(i,j) - \alpha} \frac{t_j - \mathrm{DLTADA}/V(j) + \mathrm{EPSILN}/V(i)}{\mathrm{DDSR}(n,k)} f(t_j)dt_j$$

$$+ \int_{\mathrm{DDSR}(n,k) + \overline{\mathrm{TAA}}(i,j) - \alpha}^{\infty} \frac{\mathrm{DDSR}(n,k) + \mathrm{EPSILN}/V(i) - \mathrm{DLTADA}/V(i)}{\mathrm{DDSR}(n,k)} f(t_j)dt_j \qquad (35b)$$

where
$$\alpha = \overline{\mathrm{TAA}}(i,j) + \mathrm{DLTADA}/V(i) - \mathrm{DLTADA}/V(j) \qquad (36)$$
$$\beta = \overline{\mathrm{TAA}}(i,j) + \mathrm{EPSILN}/V(i) - \mathrm{DLTADA}/V(j) \qquad (37)$$

$$\mathrm{P1A2}(n,ijk) = \int_{\overline{\mathrm{TAA}}(i,j) - \beta}^{\mathrm{DDSR}(n,k) + \overline{\mathrm{TAA}}(i,j) - \alpha} \frac{t_j}{\mathrm{DDSR}(n,k)} f(t_j)dt_j$$

$$+ \left[\frac{-\mathrm{DLTADA}/V(j) + \mathrm{EPSILN}/V(i)}{\mathrm{DDSR}(n,k)}\right] \left\{ \Phi\left[\frac{\mathrm{DDSR}(n,k) - \alpha}{\mathrm{SIGMAA}}\right] - \Phi\left[\frac{-\beta}{\mathrm{SIGMAA}}\right] \right\}$$

$$+ \left[\frac{\mathrm{DDSR}(n,k) - \mathrm{TND}(i)}{\mathrm{DDSR}(n,k)}\right] \left\{ 1 - \Phi\left[\frac{\mathrm{DDSR}(n,k) - \alpha}{\mathrm{SIGMAA}}\right] \right\} \qquad (35c)$$

where $\phi[x]$ is the standard form of the cumulative distribution function of the normal distribution:

$$\int f(x)dx = \phi\left[\frac{x - \bar{x}}{\sigma}\right]. \tag{38}$$

Evaluating the remaining integral expression is simplified by using the standarized variable z, where

$$z = (t_j - \overline{TAA}(i,j))/SIGMAA, \text{ and } dz = dt_j/SIGMAA.$$

$$\int_{\overline{TAA}(i,j)-\beta}^{DDSR(n,k)+\overline{TAA}(i,j)-\alpha} \frac{t_j}{DDSR(n,k)} f(t_j)dt_j = \frac{1}{DDSR(n,k)} \int_{\overline{TAA}(i,j)-\beta}^{DDSR(n,k)+\overline{TAA}(i,j)-\alpha} \frac{t_j}{SIGMAA\sqrt{2\pi}} e^{\frac{-(t_j-\overline{TAA}(i,j))^2}{2\,SIGMAA^2}} dt_j \tag{39a}$$

$$= \frac{1}{DDSR(n,k)} \int_{-\beta/SIGMAA}^{\frac{DDSR(n,k)-\alpha}{SIGMAA}} \frac{(SIGMAA * z + \overline{TAA}(i,j))}{SIGMAA\sqrt{2\pi}} e^{-z^2/2} * SIGMAA\, dz$$

$$= \frac{\overline{TAA}(i,j)}{DDSR(n,k)} \left\{\phi\left[\frac{DDSR(n,k)-\alpha}{SIGMAA}\right] - \phi\left[\frac{-\beta}{SIGMAA}\right]\right\}$$

$$+ \frac{SIGMAA}{DDSR(n,k)\sqrt{2\pi}}\left[E2 - E1\right] \tag{39b}$$

$$\text{where } E2 = e^{-\beta^2/2\,SIGMAA^2} \tag{40}$$

$$E1 = e^{-(DDSR(n,k)-\alpha)^2/2\,SIGMAA^2}. \tag{41}$$

We can now state the complete expression for P1A2. First, however, we note that:

$$P1(ijk) = \Phi\left[\frac{\overline{TAA}(ij) + EPSILN/V(i) - DLTADA/V(j)}{SIGMAA}\right] = \Phi\left[\frac{\beta}{SIGMAA}\right] \quad (42a)$$

and therefore

$$\Phi\left[\frac{-\beta}{SIGMAA}\right] = 1. - P1(ijk). \quad (42b)$$

Also, we will define

$$PT1(n,ijk) = \Phi\left[\frac{DDSR(n,k) - \alpha}{SIGMAA}\right]. \quad (43)$$

This yields:

$$\begin{aligned} P1A2(n,ijk) = {} & \frac{\overline{TAA}(i,j)}{DDSR(n,k)} \left[PT1(n,ijk) - (1. - P1(ijk))\right] \\ & + \frac{SIGMAA}{DDSR(n,k)\sqrt{2\pi}} \left[E2 - E1\right] \\ & + \frac{(-DLTADA/V(j) + EPSILN/V(i))}{DDSR(n,k)} \left[PT1(n,ijk) - (1 - P1(ijk))\right] \\ & + \left(1 - \frac{TND(i)}{DDSR(n,k)}\right) \left[1 - PT1(n,ijk)\right] \end{aligned} \quad (44a)$$

$$P1A2(n,ijk) = \frac{\beta}{DDSR(n,k)} \left[PT1(n,ijk) + P1(ijk) - 1 \right]$$

$$+ (PNE - 1)\ (PT1(n,ijk) - 1)$$

$$+ \frac{SIGMAA}{DD(n,k)\sqrt{2\pi}} \left[E2 - E1 \right]. \qquad (44b)$$

C.1.3 Combination of Terms

It is now possible to combine terms, to derive an expression for P1*(ijk). As previously stated,

$$P1^*(ijk) = P1(ijk) \left[1 - \sum_n Q(n,i) \right] + \sum_n \left[P1A(n,ijk)^* \ Q(n,i) \right] \qquad (29a)$$

which can now be evaluated as

$$\left. \begin{aligned} P1^*(ijk) &= P1(ijk) + \sum_n Q(n,i)^* \ (-P1(ijk) + P1(ijk)) \\ &= P1(ijk) \end{aligned} \right\} \text{if } DDSR(n,k) \leq TND(i) \qquad (45a)$$

$$\left. \begin{aligned} = P1(ijk) + \sum_n Q(n,i)^* \ \Bigg| &- P1(ijk) + PNE^* \ P1(ijk) \\ &+ \frac{\beta}{DDSR(n,k)} \left[PT1(n,ijk) + P1(ijk) - 1 \right] \\ &+ (PNE - 1)\ (PT1(n,ijk) - 1) \\ &+ \frac{SIGMAA}{DDSR(n,k)\sqrt{2\pi}} \left[E2 - E1 \right] \Bigg| \end{aligned} \right\} \text{if } DDSR(n,k) > TND(i) \qquad (45b)$$

$$P1^*(ijk) = P1(ijk) + \sum_{n'} Q(n,i) \left\{ \left[\frac{\alpha}{DDSR(n,k)} - 1 \right] \left[PT1(n,ijk) + P1(ijk) - 1 \right] + \frac{SIGMAA}{DDSR(n,k)\sqrt{2\pi}} \left[E2 - E1 \right] \right\} \quad (46)$$

where n' represents those values of n for which

$$DDSR(n,k) > TND(i).$$

C.2 Derivation of P2*(ijk,l)

The same derivation can be performed for P2*(ijk,l), the new probability of two or more departures. The result is similar to the expression obtained for P1*(ijk).

$$P2^*(ijk,l) = P2(ijk) + \sum_{n'} Q(n,i) \left\{ \left[\frac{\alpha - DDSR(k,l)}{DDSR(n,k)} - 1 \right] \left[PT2(n,ijk,l) + P2(ijk,l) - 1 \right] + \frac{SIGMAA}{DDSR(n,k)\sqrt{2\pi}} \left[e^{\frac{-[\beta - DDSR(k,l)]^2}{2\, SIGMAA^2}} - e^{\frac{-[DDSR(n,k) + DDSR(k,l) - \alpha]^2}{2\, SIGMAA^2}} \right] \right\} \quad (47)$$

$$\text{where } PT2\ (n,ijk,l) = \Phi \left[\frac{DDSR(n,k) + DDSR(k,l) - \alpha}{SIGMAA} \right]. \quad (48)$$

This can be placed in the identical form as P1* if we define two new variables:

$$\alpha' = \alpha - DDSR(k,l) \quad (49)$$

$$\beta' = \beta - DDSR(k,l). \quad (50)$$

C.3 Derivation of P3*(ijk,l,m)

The expression for P3*(ijk,l,m), the new probability of three or more departures, again is similar to that for P1* and P2*.

$$P3^*(ijk,l,m) = P3(ijk,l,m) + \sum_{n} Q(n,i)\left\{ \left[\frac{\alpha - DDSR(k,l) - DDSR(l,m)}{DDSR(n,k)} - 1. \right] \left[PT3(n,ijk,l,m) + P3(ijk,l,m) - 1 \right] + \frac{SIGMAA}{DDSR(n,k)\sqrt{2\pi}} \left[e^{-\frac{[\beta - DDSR(k,l) - DDSR(l,m)]^2}{2\,SIGMAA^2}} - e^{-\frac{[DDSR(n,k) + DDSR(k,l) + DDSR(l,m) - \alpha]^2}{2\,SIGMAA^2}} \right] \right\} \tag{51}$$

$$\text{where } PT3\,(n,ijk,l,m) = \Phi\left[\frac{DDSR(n,k) + DDSR(k,l) + DDSR(l,m) - \alpha}{SIGMAA} \right]. \tag{52}$$

Once again, this can be simplified to the form of P1* by new variables

$$\alpha'' = \alpha - DDSR(k,l) - DDSR(l,m) \tag{53}$$

$$\beta'' = \beta - DDSR(k,l) - DDSR(l,m). \tag{54}$$

APPENDIX D

DEFAULT AND K-MODELS FOR ALL MODELS

The tables on the following pages detail the default and K-models (for calculating arrival-priority and departure-priority capacities, respectively), for all models in the current version of the FAA Airfield Capacity Program.

TABLE D-1

SINGLE RUNWAY -- SUBROUTINE SINGLE

MODEL	WEATHER	DEFAULT	K-MODEL
1-1 (A)	all	primary equation	-
1-2 (D)	all	primary equation	-
1-3 (B)	all	1-1(A) plus interleaved departures	1-2(D)
1-4* (B)		1-3(B) with predetermined interarrival times	N/A

*new model, only called by THREPA

TABLE D-2

TWO PARALLEL RUNWAYS -- SUBROUTINE TWOPA

MODEL	WEATHER	DEFAULT	K-MODEL
2-1 (F:A,A)	all	1-1(A) + 1-1(A)	-
2-2 (F:A,D)	all	1-1(A) + 1-2(D)	1-2(D)
2-3 (F:D,D)	all	1-2(D) + 1-2(D)	-
2-4 (F:B,A)	all	1-3(B) + 1-1(A)	2-2(F:A2,D1)
2-5 (F:B,D)	all	1-3(B) + 1-2(D)	2-3(F:D,D)
2-6 (F:B,B)	all	1-3(B) + 1-3(B)	2-3(F:D,D)
2-7 (M:A,A)	VMC	1-1(A) + 1-1(A)	-
	MMC/IMC	dependent arrivals, no vortex	-
2-8 (M:A,D)	all	1-1(A) + 1-2(D)	1-2(D)
2-9 (M:D,D)	all	1-2(D) + 1-2(D)	-
2-10 (M:B,A)	VMC	1-3(B) + 1-1(A)	2-8(M:A2,D1)
	MMC/IMC	2-7(M:A,A) plus departures	2-8(M:A2,D1)
2-11 (M:B,D)	all	1-3(B) + 1-2(D)	2-9(M:D,D)
2-12 (M:B,B)	VMC	1-3(B) + 1-3(B)	2-9(M:D,D)
	MMC/IMC	2-7(M:A,A) plus departures on both	2-9(M:D,D)
2-13 (N:A,A)		see 2-7(M:A,A)	
2-14 (N:A,D)		see 2-8(M:A,D)	

TABLE D-2
(Concluded)

MODEL	WEATHER	DEFAULT	K-MODEL
2-15 (N:D,D)		see 2-9(M:D,D)	
2-16 (N:B,A)		see 2-10(M:B,A)	
2-17 (N:B,D)		see 2-11(M:B,D)	
2-18 (N:B,B)		see 2-12(M:B,B)	
2-19 (C:A,A)	VMC	independent arrivals with vortex	-
	MMC/IMC	1-1(A1) with average mix	-
2-20 (C:A,D)	VMC/MMC	1-1(A) + 1-2(D)	1-2(D)
	IMC	dual-lane primary equation--DUAL(1)	1-2(D)
2-21 (C:D,D)	VMC	independent departures with vortex	-
	MMC/IMC	1-2(D1) with average mix	-
2-22 (C:B,A)	VMC	1-3(B) + 1-1(A) with vortex	2-20(C:A2,D1)
	MMC	1-1(A2) + 1-2(D1)	1-2(D)
	IMC	2-20(C:A2,D1)	1-2(D)
2-23 (C:B,D)	VMC	1-3(B) + 1-2(D) with vortex	2-21(C:D,D)
	MMC	1-1(A) + 1-2 (D)	1-2(D)
	IMC	2-20(C:A,D)	1-2(D)
2-24 (C:B,B)	VMC	1-3(B) + 1-3(B) with vortex	2-21(C:D,D)
	MMC	1-1(A) + 1-2 (D)	1-2(D)
	IMC	2-20(C:A,D)	1-2(D)

TABLE D-3

THREE PARALLEL RUNWAYS -- SUBROUTINE THREPA

MODEL		WEATHER	DEFAULT	K-MODEL
3-1	(C:B,B,B)	VMC	1-3(B) + 1-3(B) + 1-3(B) with vortex	3-28(C:D,D,D)
		MMC	2-12(M:B1,B3)	3-22(C:A,D,D)
		IMC	2-12(M:B1,B3)	2-9(M:D1,D3)
3-2	(M:B,B,B)	VMC	2-24(C:B1,B2) + 1-3(B3)	3-29(M:D,D,D)
		MMC	1-2(D2) + 2-10(M:B3,A1)	3-23(M:A,D,D)
		IMC	3-5(M:A,D,B)	3-29(M:D,D,D)
3-3	(M:B,B,B)	VMC	2-24(C:B1,B2) + 1-3(B3)	3-29(M:D,D,D)
		MMC	2-20(C:A,D) + 1-3(B3)	3-24(F:A,D,D)
		IMC	2-20(C:A,D) + 1-3(B3)	3-29(M:D,D,D)
3-4	(C:A,D,B)	VMC	1-1(A) + 2-23(C:B3,D2)	3-22(C:A,D,D)
		MMC	1-2(D2) + 2-7(M:A1,A3)	3-22(C:A,D,D)
		IMC	3-17(C:A,D,A)	2-21(C:D2,D3)
3-5	(M:A,D,B)	VMC	1-1(A) + 1-2(D) + 1-3(B)	3-23(M:A,D,D)
		MMC	2-10(M:B3,A1) + 1-2(D2)	3-23(M:A,D,D)
		IMC	2-7(M:A1,A3) + D3 from 1-4(B3) + D2 from DUAL(2,A,D)*	2-9(M:D2,D3)
3-6	(F:A,D,B)	VMC/MMC	2-20(C:A,D) + 1-3(B)	3-24(F:A,D,D)
		IMC	2-20(C:A,D) + 1-3(B)	2-3(F:D2,D3)
3-7	(M:B,B,A)	VMC	2-24(C:B,B) + 1-1(A3)	3-22(C:A3,D2,D1)
		MMC	2-7(M:A1,A3) + 1-2(D2)	3-22(C:A3,D2,D1)
		IMC	2-7(M:A1,A3) + D2 from DUAL(2,A,D)*	3-22(C:A3,D2,D1)
3-8	(F:B,B,A)	VMC	2-24(C:B,B) + 1-1(A3)	3-22(C:A3,D2,D1)
		MMC/IMC	2-20(C:A,D) + 1-1(A3)	3-22(C:A3,D2.D1)

*new model

TABLE D-3
(Continued)

MODEL	WEATHER	DEFAULT	K-MODEL
3-9 (M:A,B,B)	VMC	2-22(C:B2,A1) + 1-3(B3)	3-23(M:A,D,D)
	MMC	2-7(M:A1,A3) + 1-2(D2)	3-23(M:A,D,D)
	IMC	3-5(M:A,D,B)	2-9(M:D2,D3)
3-10 (F:A,B,B)	VMC	2-22(C:A,D) + 1-3(B3)	3-24(F:A,D,D)
	MMC/IMC	2-20(C:A,D) + 1-3(B3)	2-3(F:D2,D3)
3-11 (M:B,A,D)	VMC	2-22(C:B,A) + 1-2(D3)	3-26(M:D,A,D)
	MMC/IMC	2-22(C:B,A) + 1-2(D3)	2-3(F:D1,D3)
3-12 (F:B,A,D)	VMC	2-22(C:B,A) + 1-2(D3)	3-27(F:D,A,D)
	MMC/IMC	2-22(C:B,A) + 1-2(D3)	2-3(F:D1,D3)
3-13 (M:D,A,B)	VMC	2-20(C:A2,D1) + 1-3(B3)	3-26(M:D,A,D)
	MMC	2-10(M:B3,A2) + 1-2(D1)	3-26(M:D,A,D)
	IMC	3-5(M:A2,D1,B3)	2-3(F:D1,D3)
3-14 (F:D,A,B)	VMC/MMC	2-20(C:A2,D1) + 1-3(B3)	3-27(F:D,A,D)
	IMC	2-20(C:A2,D1) + 1-3(B3)	2-3(F:D1,D3)
3-15 (M:A,D,A)	VMC/MMC	2-20(C:A,D) + 1-1(A3)	-
	IMC	2-20(C:A,D) + 1-1(A3)	2-8(M:A3,D2)
3-16 (F:A,D,A)	VMC/MMC	2-20(C:A,D) + 1-1(A3)	-
	IMC	2-20(C:A,D) + 1-1(A3)	2-2(F:A3,D2)
3-17 (C:A,D,A)	VMC	2-20(C:A,D) + 1-1(A3)	-
	MMC	2-7(M:A1,A3) + 1-2(D2)	-
	IMC	2-7(M:A1,A3) + D2 from DUAL(3,A,D)*	1-2(D2)
3-18 (C:D,A,B)	VMC	2-22(C:B3,A2) + 1-2(D1)	3-25(C:D,A,D)
	MMC	2-9(M:D1,D3) + 1-1(A2)	-
	IMC	3-25(C:D,A,D)	2-9(M:D1,D3)

TABLE D-3
(Concluded)

MODEL	WEATHER	DEFAULT	K-MODEL
3-19 (C:B,A,D)	all	see 3-18(C:D3,A2,B1)	
3-20 (C:A,B,B)	VMC	1-1(A1) + 1-3(B2) + 1-3(B3) with vortex	3-22(C:A,D,D)
	MMC	2-7(M:A1,A3) + 1-2(D2)	3-22(C:A,D,D)
	IMC	3-17(C:A,D,A)	3-22(C:A,D,D)
3-21 (C:B,B,A)	all	see 3-20(C:A3,B2,B1)	
3-22 (C:A,D,D)	VMC/MMC	2-21(C:D2,D3) + 1-1(A1)	-
	IMC	2-8(M:A1,D3)	2-21(C:D2,D3)
3-23 (M:A,D,D)	VMC/MMC	2-9(M:D2,D3) + 1-1(A1)	-
	IMC	2-20(C:A,D) + 1-2(D3)	2-9(M:D2,D3)
3-24 (F:A,D,D)	VMC/MMC	2-3(F:D2,D3) + 1-1(A1)	-
	IMC	2-20(C:A,D,) + 1-2(D3)	2-3(F:D2,D3)
3-25 (C:D,A,D)	VMC/MMC	2-3(F:D1,D3) + 1-1(A2)	-
	IMC	2-20(C:A2,D1) + D3 from 2-20(C:A2,D3)	2-9(M:D2,D3)
3-26 (M:D,A,D)	VMC/MMC	1-2(D1) + 1-1(A2) + 1-2(D3)	-
	IMC	2-20(C:A2,D1) + 1-2(D3)	2-3(F:D1,D3)
3-27 (F:D,A,D)	all	see 3-26(M:D,A,D)	
3-28 (C:D,D,D)	VMC	1-2(D1) + 1-2(D2) + 1-2(D3) with vortex	-
	MMC/IMC	2-9(M:D1,D3)	-
3-29 (M:D,D,D)	all	2-21(C:D,D) + 1-2(D3)	-

TABLE D-4

FOUR PARALLEL RUNWAYS -- SUBROUTINE FOURPA

	MODEL	WEATHER	DEFAULT	K-MODEL
4-1	(M:D,A,D,A)	VMC/MMC	2-20(C:A2,D1) + 2-20(C:A4,D3)	-
		IMC	2-20(C:A2,D1) + 2-20(C:A4,D3)	2-3(F:D1,D3)
4-2	(F:D,A,D,A)	all	see 4-1(M:D,A,D,A)	
4-3	(M:A,D,D,A)	VMC/MMC	2-20(C:A,D) + 2-20(C:A4,D3)	-
		IMC	2-20(C:A,D) + 2-20(C:A4,D3)	2-9(M:D2,D3)
4-4	(F:A,D,D,A)	all	see 4-3(M:A,D,D,A)	
4-5	(M:A,D,D,B)	VMC/MMC	2-20(C:A,D) + 2-23(C:B4,D3)	4-22(M/F:A,D,D,D)
		IMC	2-20(C:A,D) + 2-23(C:B4,D3)	3-29(M:D3,D4,D2)
4-6	(F:A,D,D,B)	all	see 4-5(M:A,D,D,B)	
4-7	(M:D,A,D,B)	VMC/MMC	2-20(C:A2,D1) + 2-23(C:B4,D3)	4-21(M/F:D,A,D,D)
		IMC	2-20(C:A2,D1) + 2-23(C:B4,D3)	3-29(M:D3,D4,D1)
4-8	(F:D,A,D,B)	all	see 4-7(M:D,A,D,B)	
4-9	(M:B,A,A,D)	VMC	2-22(C:B,A) + 2-20(C:A3,D4)	4-23(F:D,A,A,D)
		MMC	1-3(B1) + 2-20(C:A3,D4)	4-25(M:D,A,A,D)
		IMC	1-3(B1) + 2-20(C:A3,D4)	2-3(F:D1,D4)
4-10	(F:B,A,A,D)	all	see 4-9(M:B,A,A,D)	
4-11	(M:B,A,D,A)	VMC/MMC	2-22(C:B,A) + 2-20(C:A4,D3)	4-1(M:D,A,D,A)
		IMC	2-22(C:B,A) + 2-20(C:A4,D3)	2-3(F:D1,D3)

TABLE D-4
(Continued)

MODEL	WEATHER	DEFAULT	K-MODEL
4-12 (F:B,A,D,A)		see 4-11(M:B,A,D,A)	
4-13 (M:B,B,A,D)	VMC/MMC	2-24(C:A,D) + 2-20(C:A3,D4)	4-21(M/F:D4,A3,Ds,D1)
	IMC	2-24(C:A,D) + 2-20(C:A3,D4)	3-29(M:D1,D2,D4)
4-14 (F:B,B,A,D)	all	see 4-13(M:B,B,A,D)	
4-15 (M:B,A,D,B)	VMC/MMC	2-22(C:B,A) + 2-23(C:B4,D3)	4-21(M/F:D,A,D,D)
	IMC	2-22(C:B,A) + 2-23(C:B4,D3)	3-29(M:D3,D4,D1)
4-16 (F:B,A,D,B)	all	see 4-15(M:B,A,D,B)	
4-17 (M:B,B,A,B)	VMC/MMC	2-24(C:B,B) + 2-22(C:B4,A3)	4-21(M/F:D4,A3,D2,D1)
	IMC	2-24(C:B,B) + 2-22(C:B4,A3)	3-29(M:D1,D2,D4)
4-18 (F:B,B,A,B)	all	see 4-17(M:B,B,A,B)	
4-19 (M:B,B,B,B)	all	2-24(C:B,B) + 2-24(C:B3,B4)	4-24(M/F:D,D,D,D)
4-20 (F:B,B,B,B)	all	see 4-19(M:B,B,B,B)	
4-21 (M/F:D,A,D,D)	VMC/MMC	2-20(C:A2,D1) + 2-21(C:D3,D4)	-
	IMC	2-20(C:A2,D1) + 2-21(C:D3,D4)	3-29(M:D3,D4,D1)
4-22 (M/F:A,D,D,D)	VMC/MMC	2-20(C:A,D) + 2-21(C:D3,D4)	-
	IMC	2-20(C:A,D) + 2-21(C:D3,D4)	3-29(M:D3,D4,D2)

TABLE D-4
(Concluded)

MODEL	WEATHER	DEFAULT	K-MODEL
4-23 (F:D,A,A,D)	VMC/MMC	2-20(C:A2,D1) + 2-20(C:A3,D4)	-
	IMC	2-20(C:A2,D1) + 2-20(C:A3,D4)	2-3(F:D1,D4)
4-24 (M/F:D,D,D,D)	all	2-21(C:D,D) + 2-21(C:D3,D4)	-
4-25 (M:D,A,A,D)	VMC	2-20(C:A2,D1) + 2-20(C:A3,D4)	-
	MMC	1-2(D) + 2-7 (M:A2,A3) + 1-2(D4)	-
	IMC	2-7(M:A2,A3) + D1 from DUAL(2, A2,D1) + D4 from DUAL(2,A3,D4)*	2-3(F:D1,D4)

*new model

TABLE D-5

TWO OPEN-V RUNWAYS -- SUBROUTINE OPENV2

MODEL	WEATHER	IR	DEFAULT	K-MODEL
5-1 (DV*:D,D)	all	0	2-21(C:D,D)	-
		>0	1-2(D1) + 1-2(D2)	-
5-2 (DV:A,D)	VMC/MMC	all	1-1(A1) + 1-2(D2)	-
	IMC	0	2-20(C:A,D)	1-2(D2)
		>0	1-1(A1) + 1-2(D2)	-
5-3 (DV:B,D)	VMC	0	2-23(C:B,D)	5-1(DV:D,D)
		>0	1-3(B1) + 1-2(D2)	5-1(DV:D,D)
	MMC/IMC	0	2-20(C:A,D)	5-1(DV:D,D)
		>0	1-3(B1) + 1-2(D2)	5-1(DV:D,D)
5-4 (CV*:D,A)	VMC/MMC	all	1-2(D1) + 1-1(A2)	-
	IMC	0	2-20(C:A2,D1)	1-2(D1)
		1,2	1-2(D1) + 1-1(A2)	-
		3	6-2(A2,D1)	1-2(D1)
5-5 (CV:B,A)	VMC	0	2-22(C:B,A)	5-4(CV:D,A)
		>0	1-3(B1) + 1-1(A2)	5-4(CV:D,A)
	MMC	0,3	1-2(D1) + 1-1(A2)	-
		1	2-10(M:B,A)	1-2(D1)
		2	1-3(B1) + 1-1(A2)	5-4(CV:D,A)
	IMC	0	2-20(C:A2,D1)	1-2(D1)
		1	2-10(M:B,A)	5-4(CV:D,A)
		2	1-3(B1) + 1-1(A2)	5-4(CV:D,A)
		3	6-2(A,D)	1-2(D1)

*DV - diverging
CV - converging

TABLE D-6

TWO INTERSECTING RUNWAYS -- SUBROUTINE TWOIN

MODEL	WEATHER	DEFAULT	K-MODEL
6-1 (D,D)	VMC/MMC	max [primary equation, 1-2(D) with average mix]	-
	IMC	1-2(D) with average mix	-
6-2 (A,D)	all	primary equation	1-2(D2)
6-3 (B,D)	all	max [6-2(A,D), 1-3(B)]	6-1(D,D)

TABLE D-7

THREE INTERSECTING RUNWAYS -- SUBROUTINE THREIN

MODEL	WEATHER	DEFAULT	K-MODEL
7-1 (C:A,D,D)	VMC	2-20 (C:A,D)	6-1(D2,D3)
	MMC/IMC	max [2-20(C:A,D), 6-2(A,D)]	6-1(D2,D3)
7-2 (M:A,D,D)	all	2-8(M:A,D)	6-1(D2,D3)
7-3 (C:B,B,D)	VMC	2-24(C:B,B)	2-21(C:D,D)
	MMC/IMC	max [2-20(C:A,D), 5-2(A,D)]	6-1(D2,D3)
7-4 (M:B,B,D)	all	2-12(M:B,B)	2-9(M:D,D)

TABLE D-8

THREE OPEN-V RUNWAYS -- SUBROUTINE OPENV3

MODEL	WEATHER	IR	DEFAULT	K-MODEL
10-1 (DV*:B,A,D)	VMC	all	2-22(C:B,A) + 1-2(D3)	10-5(DV:D,A,D)
	MMC/IMC	0	3-25(C:D,A,D)	2-9(M:D1,D3)
		>0	2-20(C:A,D) + 1-2(D3)	2-9(M:D1,D3)
10-2 (DV:B,B,D)	VMC	0	2-22(C:B,A) + 1-2(D3)	3-28(C:D,D,D)
		>0	2-24(C:B,B) + 1-2(D3)	3-29(M:D,D,D)
	MMC/IMC	all	see 10-1(DV:B,A,D)	
10-3 (CV*:B,D,A)	VMC	all	2-23(C:B,D) + 1-1(A3)	3-22(C:A3,D2,D1)
	MMC	0	2-7(M:A1,A3) + 1-2(D2)	2-21(C:D,D)
		1,2	d < 3500: 2-7(M:A1,A3) + 1-2(D2)	3-22(C:A3,D2,D1)
			d ≥ 3500: 1-1(A1) + 1-2(D2) + 1-1(A3)	3-22(C:A3,D2,D1)
		3	2-21(C:D,D) + 1-1(A3)	-
	IMC	0	3-4(C:A,D,A)	2-21(C:D,D)
		1,2	d < 3500: 3-7 (M:A,D,A)	2-21(C:D,D)
			d ≥ 3500: 2-20(C:A,D) + 1-1(A3)	2-21(C:D,D)
		3	2-20(C:A,D)	2-21(C:D,D)
10-4 (CV:B,B,A)	VMC	all	2-24(C:B,B) + 1-1(A3)	3-22(C:A3,D2,D1)
	MMC/IMC	all	see 10-3(CV:B,D,A)	
10-5 (DV:D,A,D)	VMC	all	2-20(C:A2,D1) + 1-2(D3)	-
	MMC/IMC	all	see 10-1(DV:B,A,D)	

*DV - diverging
CV - converging

TABLE D-9

FOUR OPEN-V RUNWAYS -- SUBROUTINE OPENV4

MODEL	WEATHER	IR	DEFAULT	K-MODEL
11-1 (DV*:A,A,D,D)	VMC	all	2-19(C:A,A) + 2-21(C:D3,D4)	-
	MMC	all	2-20(C:A2,D3), average mix	2-21(C:D3,D4)
	IMC	0	2-20(C:A2,D3), average mix	2-21(C:D3,D4)
		>0	1-1(A2) + 1-2(D3), average mix	-
11-2 (DV:B,B,D,D)	VMC	0	2-22(C:B,A) + 2-21(C:D3,D4)	3-28(C:D1,D2,D4)
		>0	2-24(C:B,B) + 2-21(C:D3,D4)	4-24(M/F:D,D,D,D)
	MMC/IMC	all	2-20(C:A,D) + 2-21(C:D3,D4)	4-24(M/F:D,D,D,D)
11-3 (CV*:A,A,D,D)	VMC	all	2-19(C:A,A) + 2-21(C:D3,D4)	-
	MMC	all	2-20(C:A3,D3), average mixes	2-21(C:D3,D4)
	IMC	<3	2-19(C:A,A) + 2-21(C:D3,D4)	-
		3	6-2(A2,D3)	2-21(C:D3,D4)
11-4 (CV:B,B,D,D)	all	all	see 11-3(CV:A,A,D,D)	

*DV - diverging
CV - converging

TABLE D-10

TWO RUNWAYS INTERSECTING BEYOND -- SUBROUTINE TWOINB

MODEL	WEATHER	IR	DEFAULT	K-MODEL
12-1 (DV*:A,D)	VMC/MMC	all	1-1(A1) + 1-2(D2)	-
	IMC	0	2-20(C:A,D)	1-2(D2)
		>0	1-1(A1) + 1-2(D2)	-
12-2 (DV:B,D)	VMC	0	2-23(C:B,D)	2-21(C:D,D)
		>0	1-3(B1) + 1-2(D2)	2-9(M:D,D)
	MMC/IMC	0	2-20(C:A,D)	2-21(C:D,D)
		>0	1-3(B1) + 1-2(D2)	2-9(M:D,D)
12-3 (CV*:D,A)	VMC/MMC	all	1-2(D1) + 1-1(A2)	-
	IMC	0	2-20(C:A2,D1)	1-2(D)
		1,2	1-2(D1) + 1-1(A2)	-
		3	6-2(A2,D1)	1-2(D)
12-4 (CV:B,A)	VMC	all	5-5(CV:B,A)	12-3(CV:D,A)
	MMC/IMC	all	5-5(CV:B,A)	1-2(D)

*DV - diverging
CV - converging

TABLE D-11

THREE RUNWAYS INTERSECTING BEYOND -- SUBROUTINE THRINB

MODEL	WEATHER	IR	DEFAULT	K-MODEL
13-1 (DV*:B,A,D)	VMC	all	10-1(DV:B,A,D)	3-26(M:D,A,D)
	MMC/IMC	all	10-1(DV:B,A,D)	2-9(M:D1,D3)
13-2 (DV:B,B,D)	VMC	0	10-2(DV:B,B,D)	3-28(C:D,D,D)
		>0	10-2(DV:B,B,D)	3-29(M:D,D,D)
	MMC/IMC	0	10-2(DV:B,B,D)	2-9(M:D1,D3)
		>0	10-2(DV:B,B,D)	3-29(M:D,D,D)
13-3 (CV*:B,D,A)	VMC/MMC	all	10-3(CV:B,D,A)	3-22(C:A3,D2,D1)
	IMC	all	10-3(CV:B,D,A)	2-21(C:D,D)
13-4 (CV:B,B,A)	VMC/MMC	all	10-4(CV:B,B,A)	3-22(C:A3,D2,D1)
	IMC	all	10-4(CV:B,B,A)	2-21(C:D,D)

*DV - diverging
CV - converging

TABLE D-12

FOUR RUNWAYS INTERSECTING BEYOND -- SUBROUTINE FOUINB

MODEL	WEATHER	IR	DEFAULT	K-MODEL
14-1 (DV*:A,A,D,D)	VMC	all	11-1(DV:A,A,D,D)	-
	MMC	all	11-1(DV:A,A,D,D)	2-21(C:D3,D4)
	IMC	0	11-1(DV:A,A,D,D)	2-21(C:D3,D4)
		>0	11-1(DV:A,A,D,D)	-
14-2 (DV:B,B,D,D)	VMC	0	11-2(DV:B,B,D,D)	3-28(C:D1,D2,D4)
		>0	11-2(DV:B,B,D,D)	4-24(M/F:D,D,D,D)
	MMC/IMC	all	11-2(DV:B,B,D,D)	4-24(M/F:D,D,D,D)
14-3 (CV*:A,A,D,D)	VMC/MMC	all	11-3(CV:A,A,D,D)	-
	IMC	<3	6-2(A1,D3)	-
		3	6-2(A1,D3)	2-21(C:D3,D4)
14-4 (CV:B,B,D,D)	VMC/MMC	all	11-3(CV:A,A,D,D)	-
	IMC	<3	11-3(CV:A,A,D,D)	-
		3	11-3(CV:A,A,D,D)	2-21(C:D3,D4)

*DV - diverging
CV - converging

TABLE D-13

FOUR INTERSECTING RUNWAYS -- SUBROUTINE FOURIN

MODEL	WEATHER	DEFAULT	K-MODEL
15-1 (A,A,D,D)	VMC	2* 6-2(A2,D3) with altered inputs	2-21(C:D3,D4)
	MMC/IMC	6-2(A2,D3) with altered inputs	2-21(C:D3,D4)
15-2 (B,B,D,D)	VMC	max [2* 6-2(A2,D3) with altered inputs, 2-24 (C:B,B)]	2-21(C:D3,D4)
	MMC/IMC	max [2-20(C:A,D), 6-2(A1,D4)]	2-21(C:D,D)

TABLE D-14

DUAL LANE RUNWAY LOGIC -- SUBROUTINE DUAL

MODEL	EXPLANATION
1	Arrivals are on runway #1, departures are on #2 -- standard dual-lane -- interarrival times from subroutine MAIN
2	Alternating arrivals to runways 1 and 2 -- interarrival times are measured between consecutive arrivals to the same runway (from subroutine STAGGR)
3	Alternating arrivals to runways 1 and 2 -- interarrival times are measured between successive arrivals, either runway (from subroutine STAGGR)

APPENDIX E

REFERENCES

1. Federal Aviation Administration, "Air Traffic Control," Handbook 7110.65B, 1 January 1980.

2. Douglas Aircraft Company, et al., "Procedures for Determination of Airport Capacity -- Interim Report," January 1973, draft copy.

3. Douglas Aircraft Company, et al., "User Manual for Airfield Capacity and Delay Models -- Section I Model Description and Appendices," February 1975.

4. C. T. Ball, "Model Users' Manual for Airfield Capacity and Delay Models -- Book I," Federal Aviation Administration, FAA-RD-76-128, November 1976.

5. W. J. Swedish, "Impact of FAA E&D Elements on Miami International Airport," The MITRE Corporation, MTR-7350 Vol. VI, September 1977.

6. Dr. R. R. Iyer, "Impact of FAA E&D Elements on JFK International and LaGuardia Airports," The MITRE Corporation, MTR-7350 Vol. II, April 1977.

7. International Business Machines Corporation, "System/360 Scientific Subroutine Package, Version III, Programmer's Manual", GH20-0205-4, August 1970.

8. C. Hastings, Jr., Approximations for Digital Computers, Princeton University Press, Princeton, NJ., 1955.

9. M. Abramowitz and I. Stegun, Handbook of Mathematical Functions, National Bureau of Standards Applied Mathematics Series No. 55, U.S. Government Printing Office, June 1964.

10. A. L. Haines, "Parameters of Future ATC Systems Relating to Airport Capacity/Delay," The MITRE Corporation, MTR-7766, Rev. 1 (FAA-EM-78-8A), June 1978.

11. M. L. Schoen, et al., "Aircraft and Avionics Related Research Required to Develop an Effective High-Speed Runway Exit System," Douglas Aircraft Company, NASA Contractor Report 159075, June 1979.